

Development of a Performance Data Reduction, Analysis, and Reporting System for a  
Wind-Diesel Hybrid Power System

James Stack

Office of Science, Department of Energy ERULF Program

Bucknell University

National Renewable Energy Laboratory

Golden, Colorado

August 17, 2000

Prepared in partial fulfillment of the requirements of the Office of Science, DOE ERULF Program under the direction of Stephen Drouilhet at the National Renewable Energy Laboratory's National Wind Technology Center (NWTC).

Participant: \_\_\_\_\_

Research Advisor: \_\_\_\_\_

## **Table of Contents**

<u>Section</u>	<u>Page Number</u>
Abstract	iii.
Background	1
Control of a Wind-Diesel Hybrid Power System	3
The Data Decoding and Reduction Process	5
Data Decoding Program	7
Data Summarizing and Reporting Program	8
Acknowledgements	11
References	12
Figures	13
Appendices	26

## **Abstract**

Development of a Performance Data Reduction, Analysis, and Reporting System for a Wind-Diesel Hybrid Power System. JAMES STACK (Bucknell University, Lewisburg, Pennsylvania, 17837). Stephen Drouilhet (National Renewable Energy Laboratory, Golden, Colorado 80401).

A wind-diesel hybrid power system is by nature a complex system whose many components must be controlled at all times in order to ensure that the system not only meets its load requirements, but that the voltage and frequency of the power it produces remain within acceptable limits. Thus it is necessary to monitor the system's performance continuously to verify that the desired level of power quality is maintained, as well as to help quantify the benefits of the added wind power. System performance data is recorded by a Programmable Logic Controller (PLC), which also controls the entire system. This data is stored and sent to a web server at a remote location (the NWTC) by a BASIC Co-Processor Module with a built-in telephone modem. Two software programs were developed in order to decode and analyze the data and display it in a useful format. The first program, written in PERL, is a fully automated program that runs on a UNIX platform. When the Co-Processor sends the data it issues the command to run the program, which then decodes the data (which is sent in hexadecimal format) into text, Boolean values (ON/OFF), and decimal values, depending on the type of data. It then saves this information to a monthly text file. The second program, an EXCEL spreadsheet program, analyzes the data file and processes it to a monthly report featuring summary tables, graphs, and event logs. This monthly report can then be converted to a PDF file and copied to the Wales web site, making it available to all project stakeholders.

## **Background**

Today, much of the world's population has the benefit of living in or near a relatively urban area connected to an electric grid network with centralized generating capacity. However, there are still many parts of the world, particularly in non-industrialized or remote regions, where there is no grid connection but there still exists a need for power. In such situations, the power—whether it be for purposes of communication, irrigation, or for serving a small village community—must be supplied by some decentralized source. "At present, the most common way to supply electricity to remote loads, whether communities or special applications, is with a diesel engine driving a generator set" (Hunter, 1994). The typical diesel generator, though, while quite reliable, is usually very inefficient, expensive, and basically a poor solution to the problem. For example, after accounting for maintenance and fuel transportation costs, diesel-generated electricity can cost as much as \$.15 to \$1.00 per kWh in some areas (Drouilhet, 1999). In addition, the diesels also come with significant environmental costs including air and noise pollution and the possibility of fuel spills or leaks during transport or storage.

One promising solution to the energy problem for these remote locations is to supplement the diesel generators with renewable energy sources, such as wind power or photovoltaics. Where wind is plentiful, as it often is in some of these out-of-the-way regions, wind turbines can be an excellent source of power, drastically reducing the amount of diesel fuel consumed. In fact, one of the primary causes for the remoteness of some of these areas—particularly in extremely high latitude regions or in very mountainous regions—is the extreme weather they regularly face. While their cold, windy climates generally do not lend themselves to lucrative tourism industries, they can make these villages strong candidates for wind-diesel hybrid power systems. In many cases, simply incorporating one or two small wind turbines can provide virtually all the power a

village requires, with the diesel generators on hand primarily as a backup for occasional short periods of low wind.

The software development project presented in this paper is to be used as part of a wind-diesel power system in Wales, Alaska, a small Eskimo village of 160 people on the Bering Strait. Prior to the installation of the two AOC 66 kW wind turbines, Wales received all of its electric power from a cluster of three diesel generators (75 kW, 142 kW, and 148 kW). For several reasons, Wales was an ideal candidate for a wind-diesel hybrid system. First, as a result of its coastal location and Arctic climate, Wales has one of the best wind resources in the entire United States. In fact, the Department of Energy (DOE) has labeled this a Class 7 region (average wind speed greater than 8.8 m/s and average wind power density greater than 800 W/m<sup>2</sup>), the highest possible classification. Secondly, one of the only reasons that villagers can currently afford to rely so heavily on diesel fuel is that the state of Alaska provides a significant fuel subsidy (as high as \$0.41 per kWh) to rural residents to ease this cost burden. But due to declining oil revenues, the state is expected to eliminate this subsidy within a few years. And finally, despite Alaska's crude oil resource, the fuel the villagers use must be delivered to them by barge due to the lack of road access and the village's location on the Bering Strait. However, with the extreme cold temperatures the area experiences, most interior rivers and ports are regularly frozen over and only accessible during the summer months. As a result, large quantities of fuel must be stored on-site in bulk fuel tanks, which pose a significant threat of leaks or spills. These reasons, as well as the traditional arguments for wind energy, contribute to Wales' great potential as a site for a high-penetration wind-diesel hybrid system.

## **Control of a Wind-Diesel Hybrid Power System**

Although wind energy is clean, emitting no pollutants or greenhouse gases, and completely inexhaustible, it is also an intermittent source and so can not be relied upon to provide power on a consistent basis. Also, because the power produced by the wind is proportional to the cube of the wind speed, even small fluctuations in speed—particularly short-term (second to second) fluctuations but also long-term (season to season) fluctuations as well—can lead to substantial changes in the amount of power produced. Given the high variability of the wind, as well as the variability in the load requirements of the typical village, a relatively sophisticated control system is usually required to effectively integrate the wind turbine with the existing diesel system. For the Wales wind-diesel system, a Programmable Logic Controller (PLC) is used for this task.

The PLC receives all system data continuously and, after entering these values in a power flow management algorithm, will dictate which components to turn on or off. Essentially, the PLC ensures that the system runs efficiently and provides high quality power at all times. For instance, when the winds are strong enough to meet the full load, the control system makes sure the diesel engines are shut off, thus saving fuel. And when one or more of the diesels have to be turned on, the control system makes sure that they are running at relatively high loads, where they operate most efficiently. The control system is also responsible for regulating the frequency and voltage of the power produced. Because no matter how strong the winds are, if the system does not provide good power quality, as measured by frequency and voltage stability, it will not be viable (Drouilhet 1999).

The primary purpose of integrating wind turbines into an existing diesel network is to reduce the amount of fuel consumed, in order to save money and reduce the environmental

impacts of the diesel generators. To maximize these savings, a short-term energy storage system can be used. The storage system, which in this case is just a battery bank, is put into use whenever the instantaneous net load (the village load minus the available wind power) is not zero. As long as the battery is not full it provides a place to store any excess power the turbines produce—which would otherwise simply go to waste or be used for some secondary application like space heating—when the instantaneous wind power exceeds the village load. It also helps to prevent unnecessary diesel starts that would ordinarily take place whenever there is even a possibility that a load peak or a transient wind power drop will cause the load to exceed the available wind power (Drouilhet, 1999). The battery in the Wales system is sized to meet the average village load for approximately ten minutes. Thus, if the wind dies down and the available wind power is expected to be less than the village load for longer than ten minutes it will be necessary to bring another diesel engine on-line.

The Wales wind-diesel system architecture consists of three diesel generators, two wind turbines, batteries, local and remote dump loads, a rotary power converter, and an auxiliary battery charter (Fig. 1). With so many mutually dependent components and the need for consistently high power quality, it is essential for every member of the system to be coordinated with the others and functioning properly at all times. It is therefore necessary to monitor each component's performance and status continuously. Since the PLC already records all of the system data for control purposes, it can be used to monitor the system as well simply by storing this information somewhere for later analysis.

## **The Data Decoding and Reduction Process**

While the PLC is capable of recording, storing, and sending information on every aspect of the entire wind-diesel system, this alone makes up only half of the overall performance monitoring process. Ultimately, when this data is uploaded to a server, it still must be decoded, analyzed, and reduced to some readable, usable form so that it is not simply an enormous pile of numbers, but actually in a form that a person can look at and easily interpret. This task is actually rather complicated, as the software that performs it is not simply an “off-the-shelf” product, but must be developed for this specific application.

When the PLC records the raw data, it transfers it to a BASIC Co-Processor module for storage. The Co-Processor, which has a built-in telephone modem, sends this data over the phone lines to a web server at the National Wind Technology Center (NWTC) in Colorado. Due to the sheer volume of data being sent, and to the slow speed of the telephone modem, the data is sent only once a day and is first converted to hexadecimal (hex) format to conserve time and space. When this data is received at the NWTC, the first step in the process is to convert each individual hex character back into its appropriate format. For instance, some of the hex characters represent text values, such as the date the information was recorded, and must be converted back to this form. Other characters represent groupings of Boolean values—essentially ON/OFF switches that indicate the status of the various components or Alarm conditions. And still others represent actual decimal values, like the wind speed or the battery voltage. But since the hexadecimal system can only be used to symbolize whole numbers, these values actually represent single precision floating point numbers in IEEE format, which in turn represent the actual decimal values.

It was decided that the software that performs this decoding operation should be written in PERL (Practical Extraction and Report Language), an interpreted high-level programming language commonly used in web site development. PERL was chosen for this task because of its powerful data manipulation abilities, as well as the fact that the program must be able to run in a UNIX environment, as that is where the raw data resides. This relatively short program first must separate the data into columns, and then read through it row by row, decoding each piece of information as it goes.

Although relatively simple in nature, this program is made slightly more complicated by the requirement that it be fully automated. When the Co-Processor sends the data each day, it also issues the command to execute the PERL program. This program must then find the appropriate monthly file to append the data to—based on the date the information was recorded—and then save the data to that file as it is decoded. The program must also be capable of creating a new monthly file and saving the data there if it detects a change in month at any point during the process.

At the end of a given month, if all goes well, the end result should be a rather lengthy text file containing an entire month's worth of ten-minute data—decoded into its appropriate format and containing all recorded information regarding the performance of the wind-diesel system. At this point, the information will basically be understandable, and, theoretically, one could sift through this mass of numbers and eventually find some meaning in it. However, due to its size, the data would still be extremely unwieldy in this form and would not be of very much use. In order to truly benefit from all of this information, another piece of software will be needed—something that can reduce this file down to a reasonable size and extract the truly important information from it. In short, something that can analyze all of this information and provide a

brief summary of it in a few charts and tables. Because this process can be accomplished most easily with a spreadsheet program, Microsoft Excel was chosen to perform for this task instead of PERL. But since Excel is not capable of running on a UNIX platform, the program, which will be a compilation of short macros written in Visual Basic for Applications (VBA), needs to live on a PC.

In essence, this program must be able to take the monthly data file and reduce it down to a neat, concise monthly report. It must be able to analyze all of the data, calculating various averages, totals, and other summary values, and then create and format several small tables with this information. Additionally, it must be able to take all of this data and generate a number of useful charts from it, showing the various trends and relationships that exist<sup>1</sup>. When this entire process is complete—when the PERL program has generated the text file of the entire month's data and the Excel program has reduced it to a neat and readable monthly report—the final step will be to make this report available on the Wales project web site. This step, which will involve converting the Excel files to Adobe Acrobat PDF files and copying them to the web site directory, will enable anyone involved in the project to access and analyze all of the system performance data at their leisure. The entire data flow, from its acquisition in the PLC to its eventual reduction to summary report for, is depicted in Figure 2.

## Data Decoding Program

The two programs, when completed, basically met all of the design criteria set for them at the beginning of the project. Although they have thus far only been used to decode and analyze simulated performance data (real data from the system is not yet available), both programs have

---

<sup>1</sup> Although it would be desirable to have this program, like the first one, be fully automated, since it will only be required to run once a month, it will likely be easiest to have the command to execute it be given manually.

performed well. The decoding program, *wales\_decoder.pl* (see Appendix A), successfully decodes every hex character into the proper format and separates each of the 98 columns, placing them under their appropriate column headers. It also checks the date on every row of data, starting a new monthly file whenever a change in month is detected. In one case, the program actually exceeded the original goals set for it. With two of the data fields, System State and System Mode, the program was originally intended to simply decode the hex character into binary form, leaving this value to be interpreted later in Excel. But in fact, the program has been written so that it can completely interpret these numbers, first decoding them to binary and then checking each of these values and printing out the associated state or mode. Altogether, the decoding program was highly successful, capable of producing a neat, orderly monthly text file of all the decoded data almost instantaneously.

## **Data Summarizing and Reporting Program**

The data summary program, on the other hand, provided a bit more of a challenge and in the end required some slight deviation from the original design plans. When the decoded text file was first imported into Excel, the neat, orderly columns of numbers produced by the PERL program became a cluttered mess, with several columns meshed together and the column headers scattered about. After careful deliberation, it was decided to import the text file first into a template program called *Monarch* in order to organize and format it into an Excel-friendly table, and then export this into Excel. Through a little experimentation it quickly became apparent what a powerful tool *Monarch* was, capable of creating new data fields calculated from the values in existing fields, breaking up binary numbers into individual bits, and splitting the data into several tables based on content. Eventually, a template was created in *Monarch* that will take each

monthly text file, add various new calculated data fields, and then split the file up into four tables—one containing all of the raw data, two containing the various alarms and Boolean values, and one containing all of the real numbers. Each of these tables is then exported to Excel where they become separate worksheets.

With the text file finally formatted and exported to Excel, the actual summary and reporting program (see Appendix B) was written without any major difficulties. This program first creates four new sheets for all of the summary information. Each of these sheets contains several summary tables, with each table representing a particular component of the system or general system or data acquisition information. There are eleven tables altogether, and they contain all of the important summary statistics for the system—averages and totals, maximums and minimums, as well as Alarm and Run time information. Examples of these tables are shown in Figure 3.

The final version of the summary program contains 20 different sub-routines that are called one at a time to produce the eleven summary tables. The first four sub-routines are used to set up and format the four worksheets containing all of the tables. The remaining 16 sub-routines are all involved in calculating the summary statistics, using the decoded data (from the original four worksheets that were imported into Excel) to generate all of the required values and then placing them in the appropriate tables. As was mentioned earlier, most of these sub-routines are designed to simply calculate averages, totals, maximums and minimums and other summary statistics from the data. But some of them go beyond this and can get rather involved. For instance, there may be times when, for whatever reason, the PLC is off or the data logging system fails. During these periods—which may last only a few minutes or may last several days—no data will be recorded or sent. Therefore, when the monthly data file is generated there

may be time segments missing from it that will not show up in the charts or tables—it will appear to be one continuous data stream. The *gaps* sub-routine was written to search out these holes in the data and to fill them in with zeros (which are colored red to set them off from the other time segments) so that the missing data segments will be evident in the charts. Another sub-routine, *missing\_segments*, counts up all of these gaps and makes a list of them, including the time they started and the time they ended, to give a better indication of the quality of the data and the reliability of the data acquisition equipment.

With the summary tables complete, the program's next task was to create the summary charts. While the creation of the tables relied almost exclusively on conventional programming techniques—many “For” loops and “If-Then” statements, for example—creating the charts involved very little of these traditional methods. Basically, the chart-making process involved extensive use Excel's "Record New Macro" feature, which enables you to record and play back actions performed in Excel, to select various data ranges and create charts from them. In Excel, the act of creating a chart is quite simple, and it turned out that generalizing this process and making it part of a macro was fairly easy. The greatest challenge in creating the charts was actually in arranging and formatting them. Splitting the charts up onto eight different sheets (usually two charts per sheet), and then changing their placements, sizes, and appearances turned out to be a rather tedious, painstaking task that required a good deal more time than actually creating them.

Once complete, the summary charts provide a nice illustration of the overall operating performance of the wind-diesel system. Fourteen different charts were created, including pie charts, wind speed histograms, daily average line graphs, time series line graphs, and power curves. Examples of these charts are shown in Figures 4 through 10. As with the summary

tables, the basic summary program contains a number of different subroutines, 24 in all, which go into the chart-making process. Again, several of these sub-routines (the first four) are involved in basic set-up activities, like creating the worksheets that the charts are placed on or calculating the monthly and daily average values that are needed in some of the charts. The next 14 are devoted to creating the actual charts, one chart per sub-routine, and basically were written by recording actions performed in Excel, as mentioned above. Finally, the last six sub-routines contain the code to arrange and format all of the charts on their individual sheets. Thus, the overall summary and reporting program contains 44 different sub-routines, divided into two basic groups (those generating tables and those generating charts), and further divided within these two areas. Because of the vast number of calculations it must perform, as well as the extensive amount of graphing involved, this program takes significantly longer to run than the data decoding program. Even with a relatively small data file this program took approximately 30 seconds to execute, and with an entire month's worth of data it could conceivably take several minutes.

## Acknowledgements

I would first like to thank the Department of Energy's Office of Science and to the National Renewable Energy Laboratory for giving me the opportunity to take part in the Energy Research Undergraduate Laboratory Fellowships program here this summer.

I would also like to offer my thanks to my two mentors, Steve Drouilhet and Mari Shirazi for their excellent guidance and support this summer. And finally, I would like to thank the entire staff here at the National Renewable Energy Laboratory, especially George Scott for his assistance in writing these programs, and to Linda Lung for her work with the ERULF program.

## References

- Drouilhet, Stephen (1999). A High-Penetration Wind-Diesel Hybrid Power System Pilot Project in Northwest Alaska. National Renewable Energy Laboratory, Golden, CO.
- Drouilhet, Stephen (1999). Power Flow Management in a High-Penetration Wind-Diesel Hybrid Power System with Short-Term Energy Storage. National Renewable Energy Laboratory, Golden, CO.
- Hunter, Ray; Eliot, George. Wind-Diesel Systems: A Guide to the Technology and its Implementation. Cambridge University Press, 1994.

## Figures

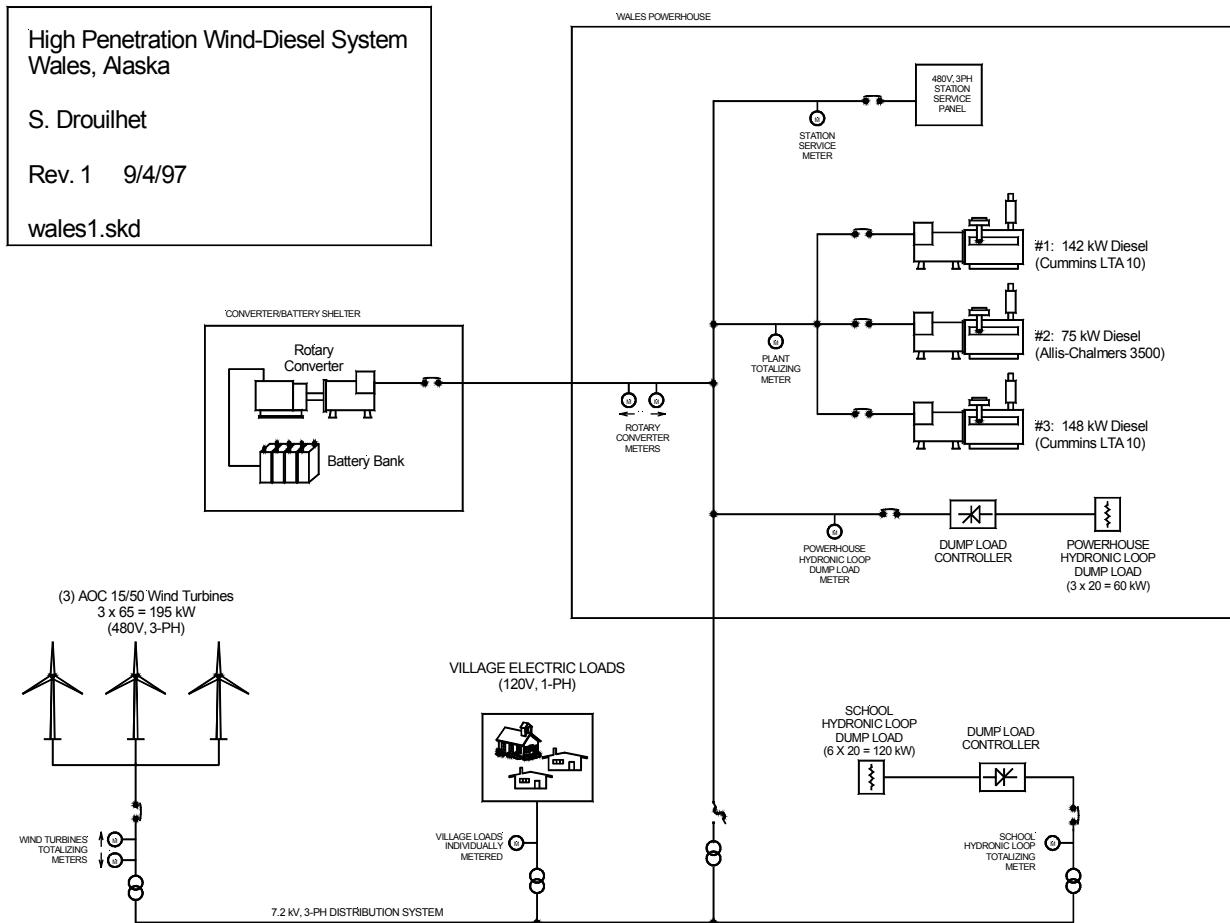


Figure 1. Wales wind-diesel hybrid system architecture

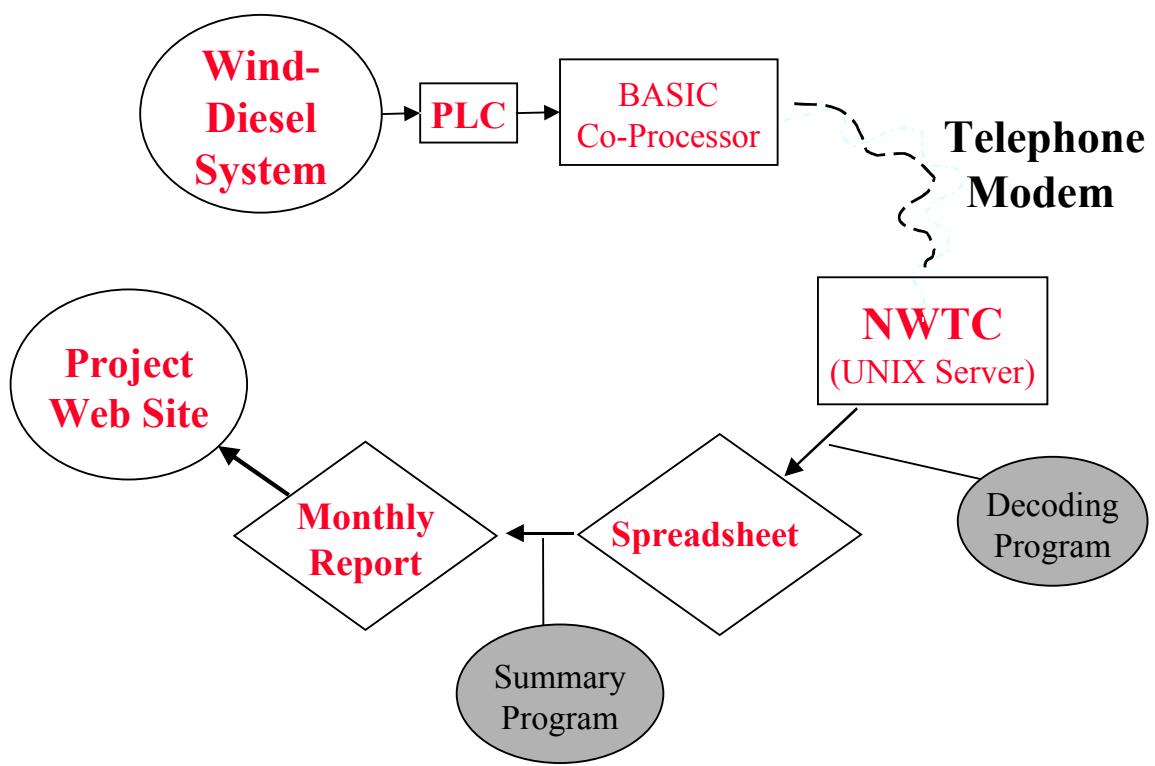


Figure 2. Data flow diagram

	Diesel 1	Diesel 2	Diesel 3	Total
Run Hours <sup>1</sup>	4.83	4.83	0.00	9.67
kWh Delivered	29.00	29.00	29.00	87.00
# Starts <sup>1</sup>	0	0	0	0
#Alarms	0	0	0	0
Alarm Time	0.00	0.00	0.00	0.00
#Warnings	0	0	0	0
Warning Time	0.00	0.00	0.00	0.00

	Wind Turb 1	Wind Turb 2	Wind Turb 3	Total
Run Hours <sup>1</sup>	4.83	0.00	0.00	4.83
kWh Delivered	29.00	29.00		58.00
#Contractor Closures <sup>1</sup>	0	0	0	0
#Alarms	0	0	0	0
Alarm Time	0.00	0.00	0.00	0.00
#Warnings	0	0	0	0
Warning Time	4.67	4.67	4.67	14.00
Capacity Factor <sup>2,3,4</sup>	0.09	0.09		0.05
Max. Inst. kW <sup>3</sup>	5.75	5.75		5.75
Avg. Hub Height Wind Speed <sup>4</sup>	5.75	5.75		---
Max. Inst. Wind Speed	5.75	5.75		---
Avg. Turbulence Intensity <sup>4,5</sup>	0.00	0.00		---

	Local Heating Load	Remote Heating Load
kWh Consumed	29.00	29.00
#Alarms	0	0
Alarm Time	0.00	0.00
#Warnings	0	0
Warning Time	0.00	0.00

	Village Load
Total kWh Consumed	29.00
Avg. kW <sup>1</sup>	5.75
Max Inst. kW	5.75
Min Inst. kW	5.75

	AC Machine	DC Machine	Rotary Converter
Run Hours <sup>1</sup>	<b>4.83</b>	<b>4.83</b>	---
RC Spin Losses (kWh)	---	---	<b>27.79</b>
#Contractor Closures <sup>1</sup>	<b>0</b>	<b>0</b>	---
#Alarms	<b>0</b>	<b>0</b>	<b>0</b>
Alarm Time	<b>4.67</b>	<b>4.67</b>	<b>4.67</b>
#Warnings	---	<b>0</b>	<b>0</b>
Warning Time	---	<b>0.00</b>	<b>4.67</b>

	Battery Bank
#Alarms	<b>0.00</b>
Alarm Time	<b>0</b>
#Warnings	<b>0.00</b>
Warning Time	<b>0.00</b>
Beginning SOC	<b>5.75</b>
Ending SOC	<b>5.75</b>
Charge kWh	<b>27.79</b>
Discharge kWh	<b>0.00</b>
Energy Losses (kWh) <sup>1</sup>	<b>27.79</b>
Energy Efficiency <sup>2</sup>	<b>0.00</b>
Charge Ah	<b>27.79</b>
Discharge Ah	<b>0.00</b>
Charge Losses (Ah) <sup>1</sup>	<b>27.79</b>
Charge Efficiency <sup>3</sup>	<b>0.00</b>
Avg. Voltage <sup>4</sup>	<b>5.75</b>
Max Inst. Voltage	<b>5.75</b>
Min Inst. Voltage	<b>5.75</b>
Avg. Temp <sup>4</sup>	<b>5.75</b>
Max Temp <sup>5</sup>	<b>5.75</b>
Min Temp <sup>5</sup>	<b>5.75</b>

	<b>ESS Shelter</b>
<b>#Alarms</b>	<b>0</b>
<b>Alarm Time</b>	<b>4.67</b>
<b>#Warnings</b>	<b>0</b>
<b>Warning Time</b>	<b>4.67</b>
Avg. Indoor Temp <sup>1</sup>	<b>5.75</b>
Max Indoor Temp <sup>2</sup>	<b>5.75</b>
Min Indoor Temp <sup>2</sup>	<b>5.75</b>
Avg. Outdoor Temp <sup>1</sup>	<b>5.75</b>
Max Outdoor Temp <sup>2</sup>	<b>5.75</b>
Min Outdoor Temp <sup>2</sup>	<b>5.75</b>

	<b>Wind Diesel Control Panel (WDCP)</b>
<b>#Alarms</b>	<b>0</b>
<b>Alarm Time</b>	<b>0.00</b>
<b>#Warnings</b>	<b>0</b>
<b>Warning Time</b>	<b>4.67</b>

<b>System Parameters</b>			
# AC Bus Alarms:	0	# AC Bus Warnings:	0
Alarm Time (hrs):	0.00	Warning Time (hrs):	0.00
<b>kWh Delivered (Wind-to-Load) with:</b>			
Diesel ON:	652.00	Diesel ON <sup>1</sup> :	1654.00
Diesel OFF:	68.00	Diesel OFF <sup>2</sup> :	6431.00
		<i>Total</i> <sup>3</sup> :	8085.00
State 0	153.00	<b>Total Wind Penetration<sup>4</sup>:</b>	
State 1A	54.00	0.78	
State 1B	22.00	<b>Max Penetration<sup>5</sup>:</b>	
State 2A	116.00	0.99	
State 2B	97.00		
State 3	10.00		
State 4	3.00		
State 5	29.00		
State 6	134.00		
State 7	64.00		
Manual Mode <sup>6</sup>	16.00		
Mode 0	345.00		
Mode 1	120.00		
Mode 2	34.00		
Mode 3	78.00		

Data	Acquisition	System
Hours in Month:	720.00	
Logged Hours:	649.00	
Missing Hours <sup>1</sup> :	71.00	

Missing	Data	Segments
Segment #	Beginning of Segment (Date & Time)	End of Segment (Date & Time)
1	6/16/00 16:43	6/16/00 17:53
2	6/16/00 18:43	6/17/00 5:13
3	6/17/00 5:53	6/17/00 13:53
4	6/17/00 14:33	6/17/00 16:33

Figure 3. Summary Report Tables

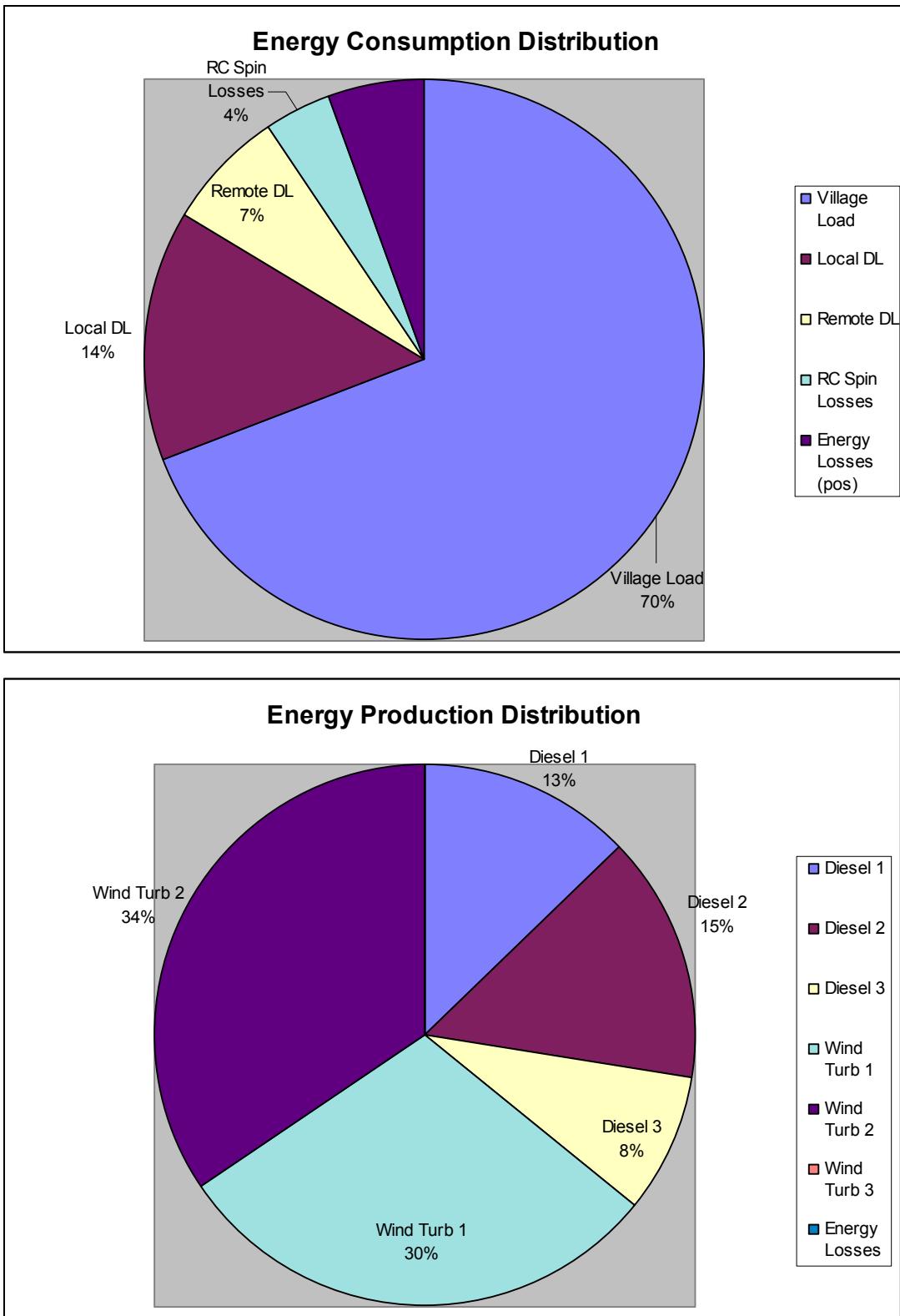


Figure 4. Summary Report pie charts

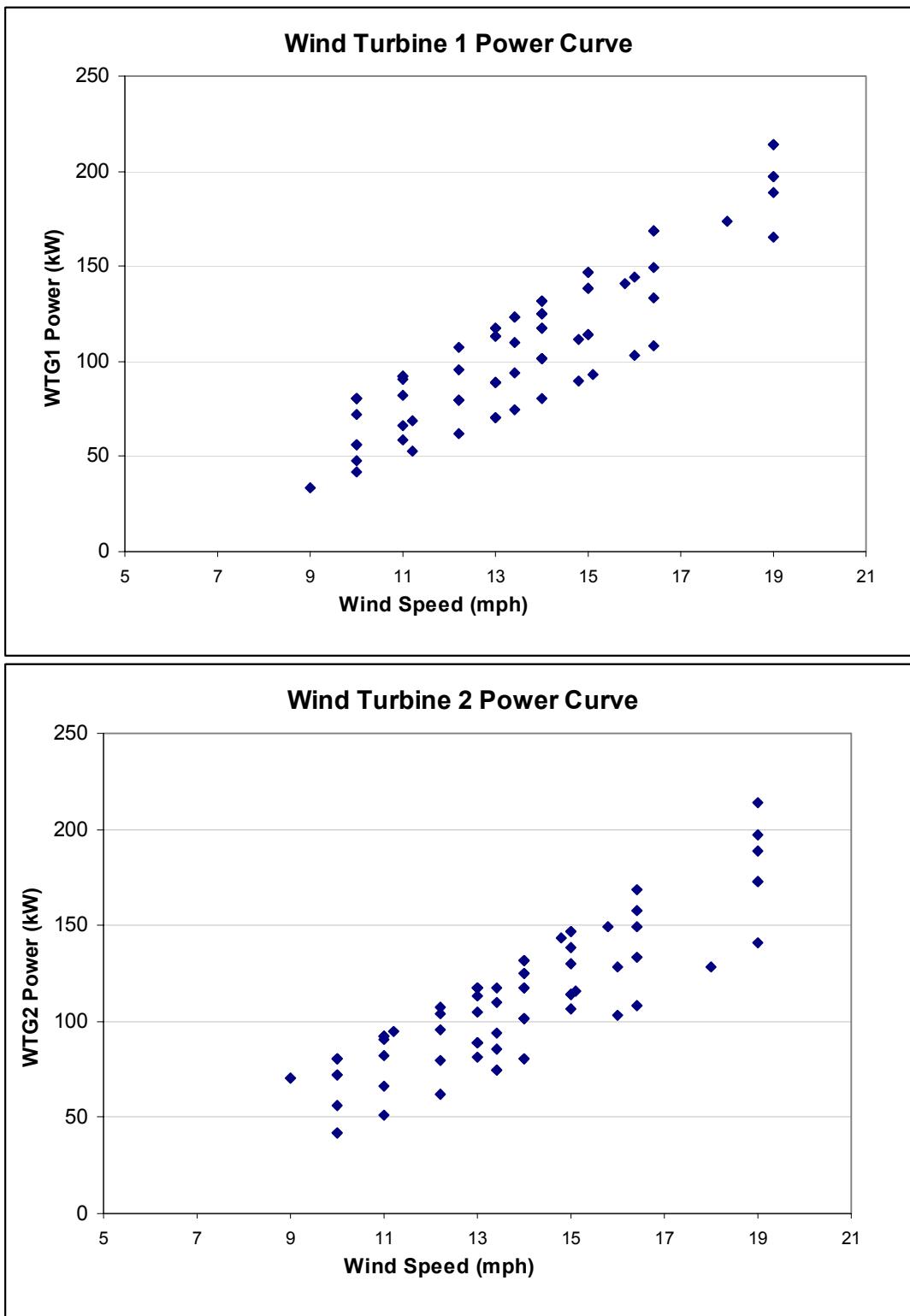


Figure 5. Summary Report power curves

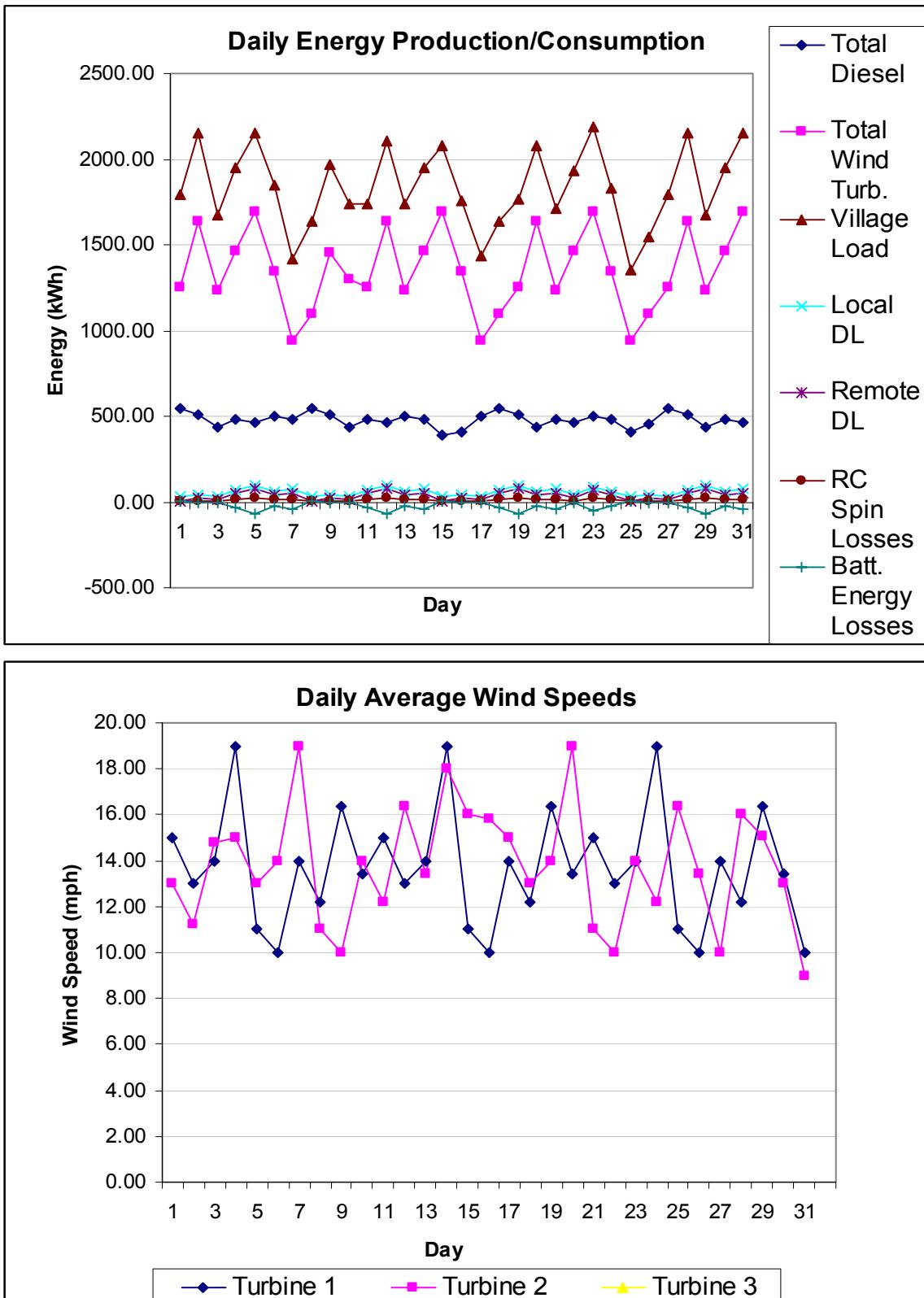


Figure 6. Summary Report Daily Average line graphs

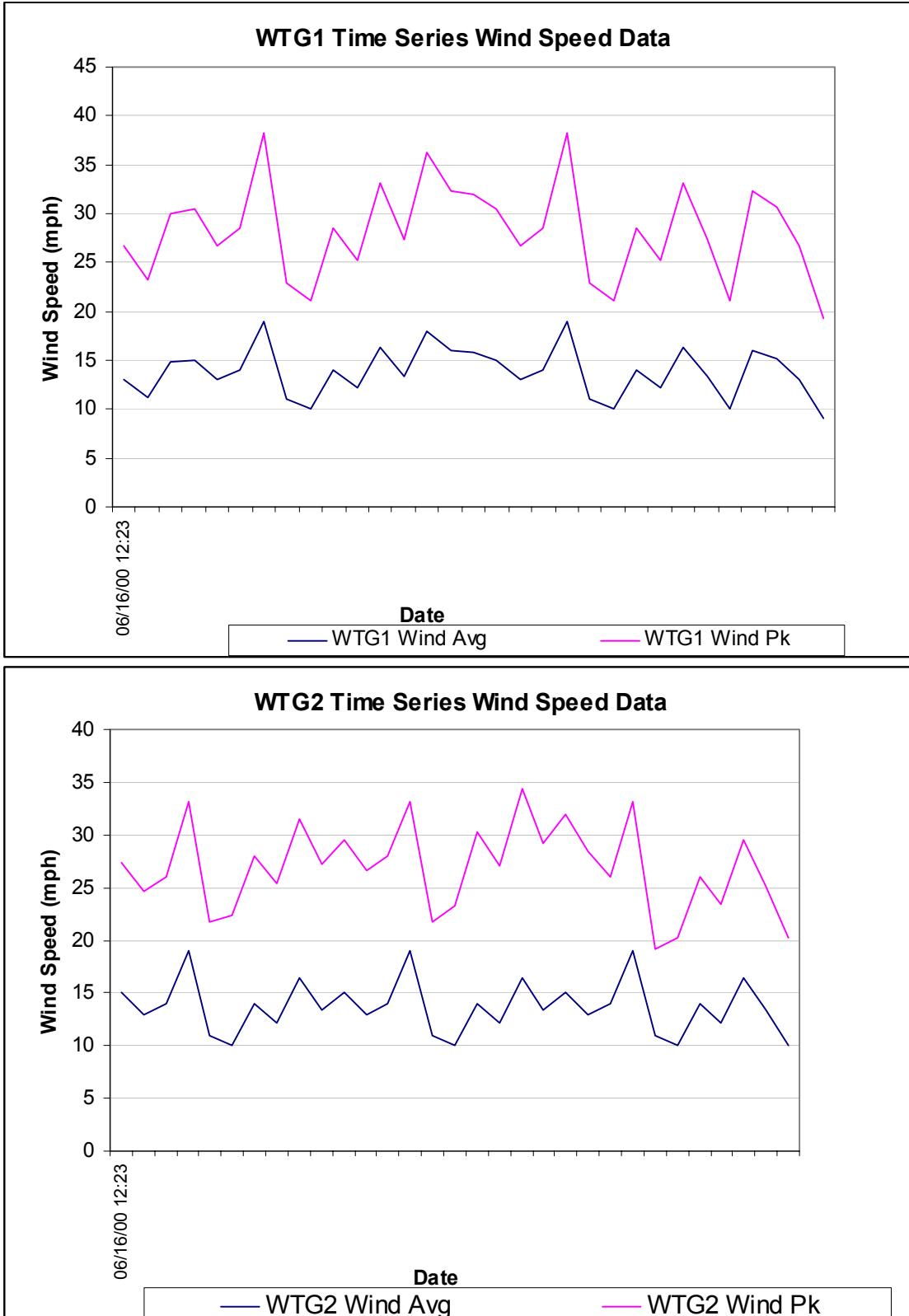


Figure 7. Summary Report Wind Speed time series charts

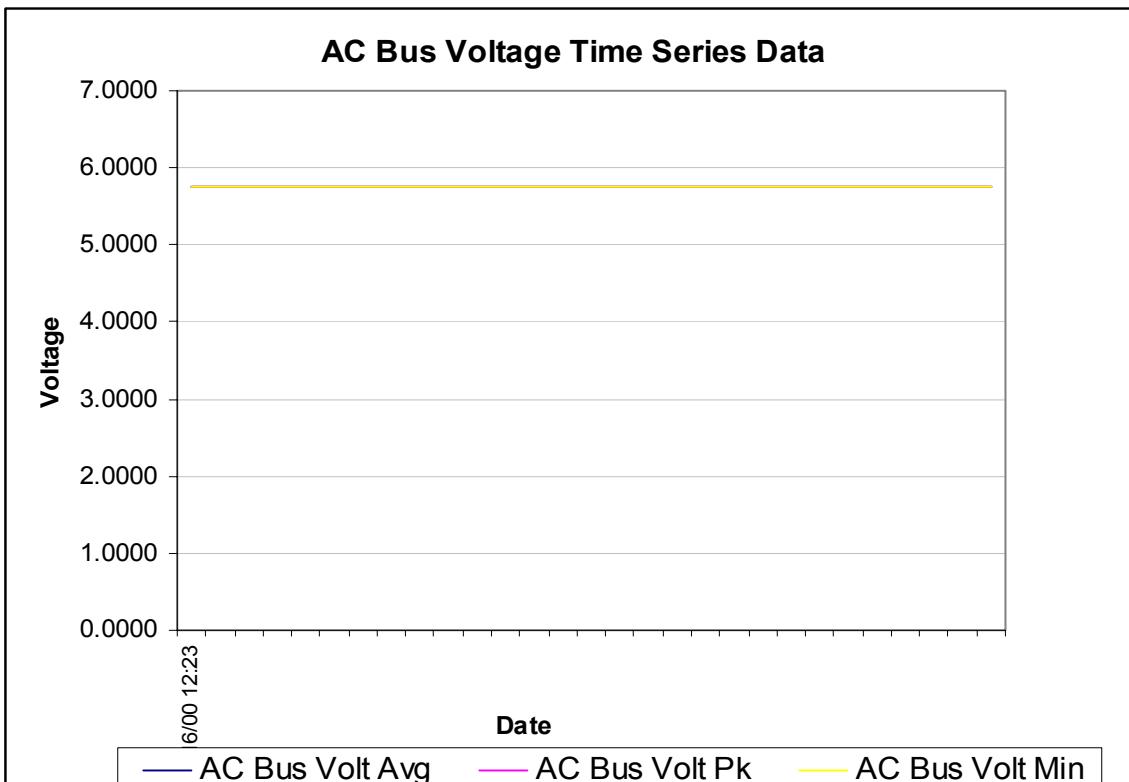
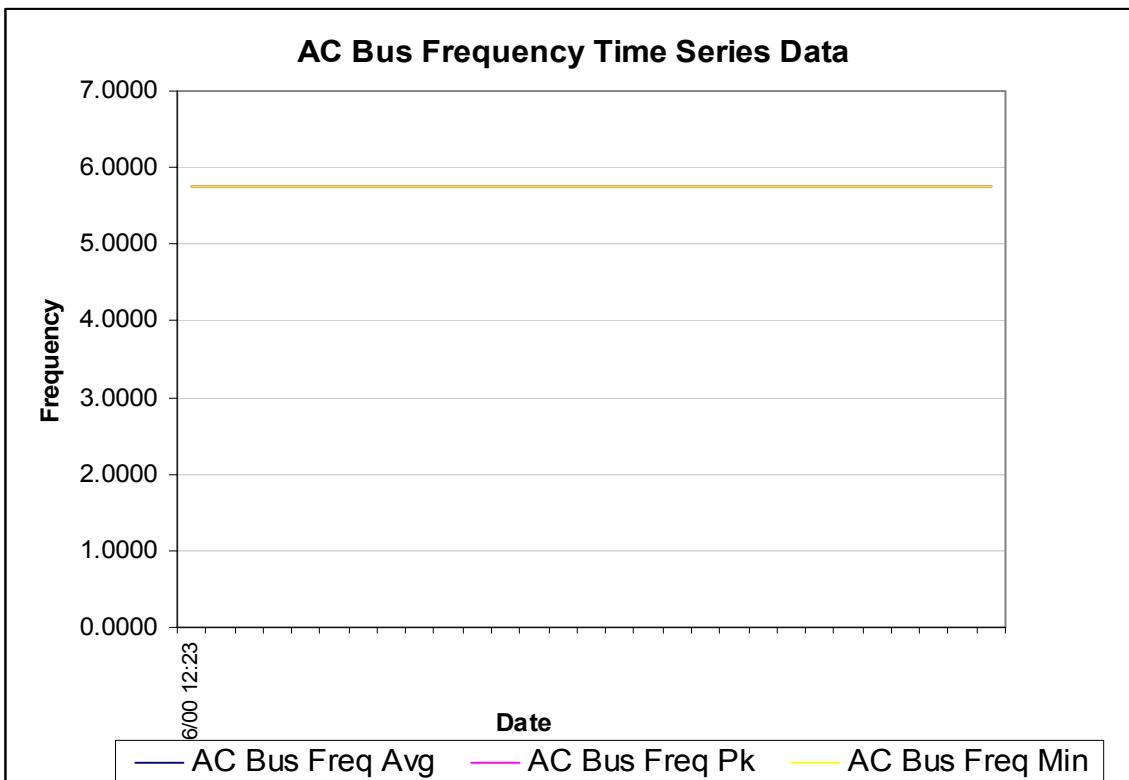


Figure 8. Summary Report Frequency and Voltage time series charts

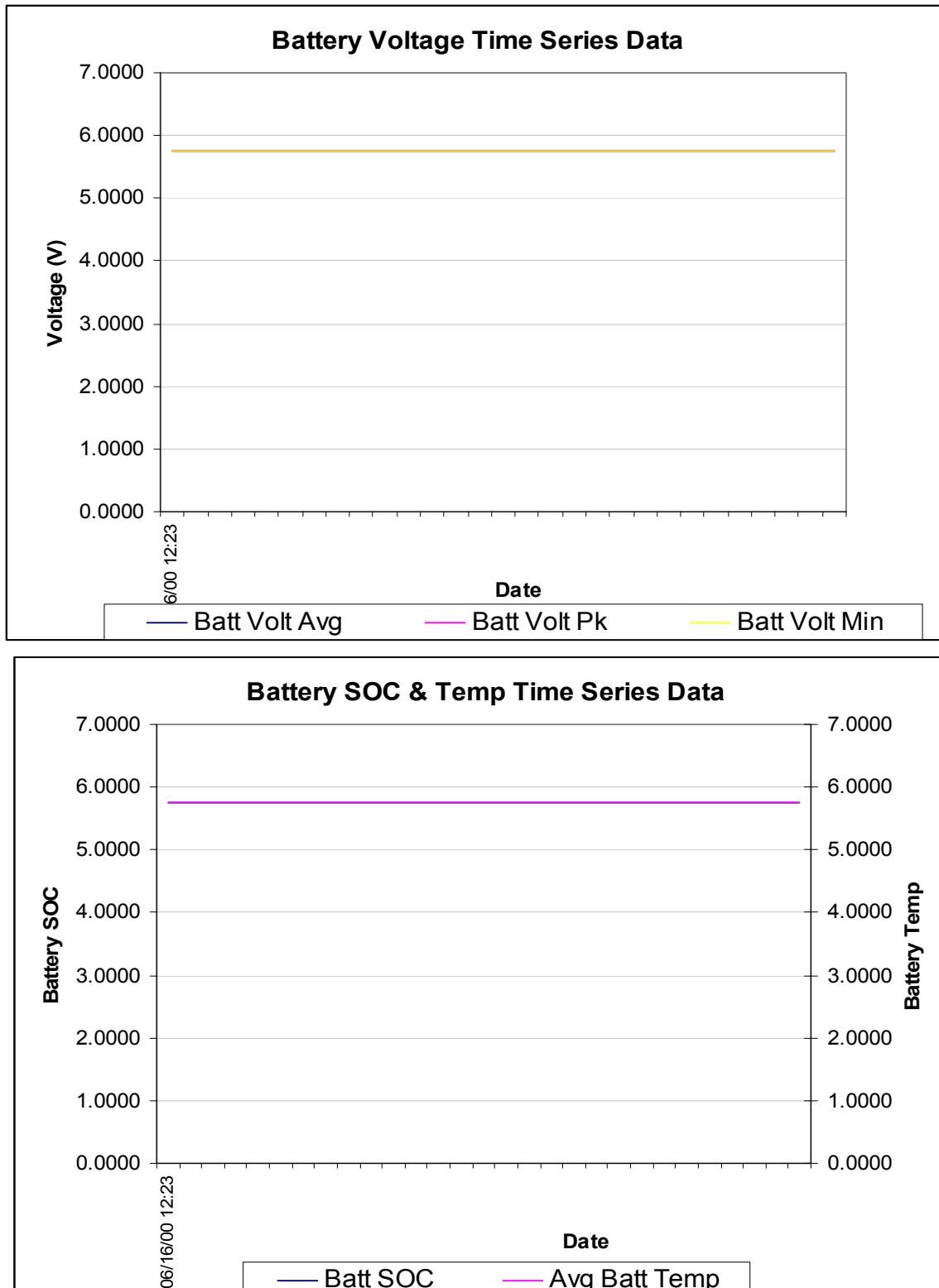


Figure 9. Summary Report Battery Voltage, State of Charge (SOC), and Temperature time series charts

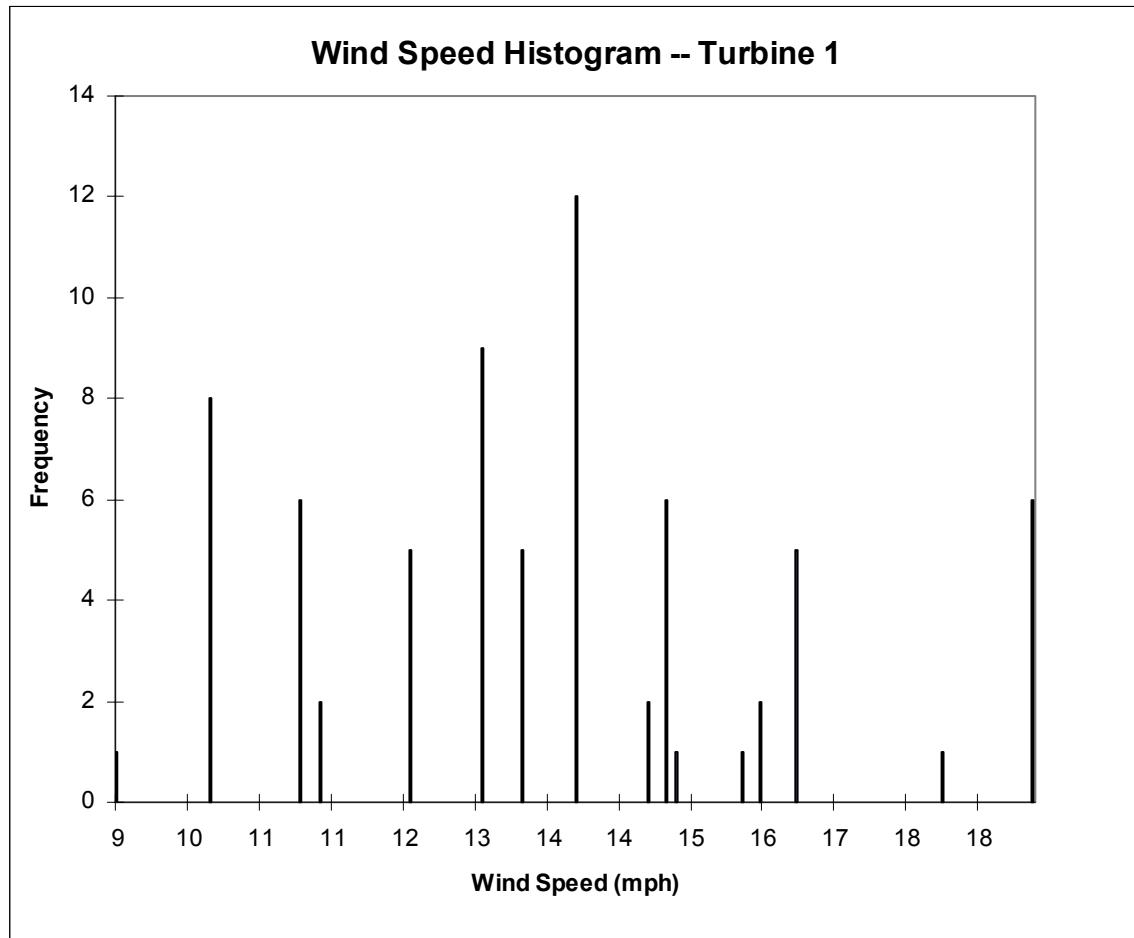


Figure 10. Summary Report Wind Speed Histogram

## Appendix A – Data Decoding Program (wales\_decoder.prl)

```
#!/bin/perl -w

#wales_decoder.prl
# The data is smooshed together so it fits in the Monarch report table.
# Headings don't match up with data in the text file produced here.
# Also keeps 16 bit numbers as 16 bits, not 2x8

#Reads and decodes data files sent from Wales, AK and appends the clean data
to a monthly
#report file (or creates a new one if it's the beginning of a month).

#Jim Stack, 6/00

#-----
-----

my ($val);
@vnames = qw (date_time system_mode system_state Diesell Diesel2
Diesel3 Wind_Turb1 Wind_Turb2 Wind_Turb3 Local_DumpLoad Remote_DumpLoad
AC_machine DC_machine Batt_charger AC_Bus_Alarms_Warnings_Cautions1A
AC_Bus_Alarms_Warnings_Cautions1B AC_Bus_Alarms_Warnings_Cautions2A
AC_Bus_Alarms_Warnings_Cautions2B WDCP_Alarms_Warnings_Cautions1A
WDCP_Alarms_Warnings_Cautions1B WDCP_Alarms_Warnings_Cautions2A
WDCP_Alarms_Warnings_Cautions2B WTG&RCCC_PLC_CommAlarmsA
WTG&RCCC_PLC_CommAlarmsB Diesell_Alarms_Warnings_CautionsA
D1_Alarms_Warnings_CautionsB Diesel2_Alarms_Warnings_CautionsA
Diesel2_Alarms_Warnings_CautionsB Diesel3_Alarms_Warnings_CautionsA
Diesel3_Alarms_Warnings_CautionsB WTG1_Alarms_Warnings_CautionsA
WTG1_Alarms_Warnings_CautionsB WTG2_Alarms_Warnings_CautionsA
WTG2_Alarms_Warnings_CautionsB WTG3_Alarms_Warnings_CautionsA
WTG3_Alarms_Warnings_CautionsB ESS_Alarms1A ESS_Alarms1B ESS_Alarms2A
ESS_Alarms2B ESS_Alarms3A ESS_Alarms3B ESS_Alarms4A ESS_Alarms4B
ESS_Alarms5A ESS_Alarms5B ESS_Warnings1A ESS_Warnings1B ESS_Warnings2A
ESS_Warnings2B ESS_Cautions1A ESS_Cautions1B ESS_Cautions2A
ESS_Cautions2B ACBusVoltAvg ACBusVoltPk ACBusVoltMinm ACBusFreqAvg
ACBusFreqPk ACBusFreqMinm D1_kW_Avg D2_kW_Avg D3_kW_Avg Dtots_kW_Avg
Dtots_kW_Pk Dtots_kW_Minm Dtots_kVAR_Avg Dtots_kVAR_Pk Dtots_kVA_Avg
Dtots_kVA_Pk WTG1_kW_Avg WTG1_kW_Pk WTG1_kW_Minm WTG1_kVAR_Avg
WTG1_Wind_Avg WTG1_Wind_Pk WTG2_kW_Avg WTG2_kW_Pk WTG2_kW_Minm
WTG2_kVAR_Avg WTG2_Wind_Avg WTG2_Wind_Pk WTGtot_kW_Avg WTGtot_kW_Pk
WTGtot_kW_Minm LocDL_kW_Avg RemoteDL_kW_Avg TotalDL_kW_Pk TotalDL_kW_Minm
SecLoad_Pk SecLoad_Minm Wnd2LdDOOn_Avg Wnd2LdDOff_Avg Vill_kW_Avg
Vill_kW_Pk Vill_kW_Minm RC_AC_kW_Avg RC_AC_kW_Pk RC_AC_kW_Minm
RC_AC_kVAR_Avg RC_AC_kVAR_Pk RC_AC_kVAR_Minm RC_AC_kVA_Avg RC_AC_kVA_Pk
RC_kWLoss_Avg Batt_Volt_Avg Batt_Volt_Pk Batt_Volt_Minm RCDCAmps_Avg
RCDCAmps_Pk RCDCAmps_Minm RCDCKW_Avg RCDCKW_Pk RCDCKW_Minm TotkVA_Ld_Pk
TotkVAR_Ld_Pk DcapReqd_Pk InstDcapReq_Pk WTcapAllwd_Min InstWTcapA_Minm
Batt_SOC Batt_Temp_Avg ESS_InsideTemp ESS_OutsideTemp);

if (@ARGV)
{ $filename = shift }
else {
    print STDERR "What file would you like to open? ";
    chomp ($filename = <STDIN>); }
```

```

open  (DATA, $filename)  or die "Can't open $filename";

$line = <DATA>;                      #Checks date on first row of data (new month or
old?)
@col = split (/,/, $line);
$mnth = hex("$col[0]");               #same as $dec[0]
@months = (January, February, March, April, May, June, July, August,
September, October, November, December);
$yr = hex("$col[2]");                 #same as $dec[2]
$datename = join("", $months[$mnth - 1], $yr);           #The name of the
file

if (-e $datename) {
    open (OUTPUT, ">$datename") or die "Can't open $datename";
}                                #If $datename exists, open and append to it
else {
    open (OUTPUT, ">$datename") or die "Can't open $datename";
    headers(); }      #If $datename doesn't exist, create it and do headers

seek DATA, 0, 0;                  #Resets <DATA> so that 1st row gets read again in
While loop

while (<DATA>)
{
    chomp;
    @row  = split (/,/);           #Separates data into seperate
                                   #columns, splitting at the
commas
    for $i (0..4) {
        $dec[$i] = hex("$row[$i]"); #Convert 1st 5 columns to
decimal
    }

    $cutdec = ($dec[2] - 2000);     #Gets last two digits of the year
    $date = sprintf "%02d/%02d/%02d", $dec[0], $dec[1], $cutdec; #Puts in
xx/xx/xx format
    $time = sprintf "%02d:%02d", $dec[3], $dec[4];           #Puts in xx:xx
24hr format

    $new_datename = join("", $months[$dec[0]-1], $dec[2]); ;
    if ($new_datename ne $datename) {
        open (OUTPUT, ">$new_datename") or die "Can't open
$new_datename";
        headers(); }

    printf OUTPUT  "%-8s  %-12s", $date, $time;

    for $l(5) {                   #System Operating Mode
        mode(); }

    for $m(6) {                   #System Operating State
        state(); }

    for $n(7) {                   #Components On-Line
        comps(); ; }
}

```

```

for $j (8..27)  {
    $bothbytes = unpack "a4", $row[$j];
    $bin = unpack("B16", pack("s", hex($bothbytes)));
    printf OUTPUT "%-22s", $bin; }

print OUTPUT "\n";

for $k (28..$#row)  {
    $fp= unpack "f",pack("i",hex("$row[$k]"));      #Convert hex to
F.P.
    printf OUTPUT "%-10.4f", $fp; }

print OUTPUT "\n";
$datename = $new_datename;
}

#-----
sub headers          #Prints column headers at the top of new monthly
files
{
#    foreach ( @vnames ) {printf OUTPUT "%-35s ", $_ ; }
#    print OUTPUT "\n" ;
#    print OUTPUT "-" x (($#vnames + 1) * 35);
#    print OUTPUT "\n" ;
}
#-----

sub mode           #For System Operating Mode column
{
my ($row);
$row = sprintf "%4s", $row[$l];
$row =~ s/\s/0/g;      #convert $row[$l] into 4-byte string with leading 0s

($hibyte,$lobyte) = unpack "a2 a2", $row;

%mode = qw ( 01 Manual_Mode    02 Mode_0     04 Mode_1    08 Mode_2    10
Mode_3 );

if ( defined ($mode {$lobyte})) {
    printf OUTPUT "%-12s", $mode{$lobyte};
}
else {
    printf OUTPUT "%-7s", "ERROR!" ;
}
}
#-----

sub state          #For System Operating State column
{
my ($row);
$row = sprintf "%4s", $row[$m];
$row =~ s/\s/0/g; # convert $row[$m] into 4-byte string with leading 0s

($hibyte,$lobyte) = unpack "a2 a2", $row;

%lo_state = qw (01 0    02 1A    04 1B    08 2A    10 2B    20 3    40 4    80 5);
%hi_state = qw (01 6    02 7);

```

```

if (defined($lo_state{$lobyte}))  {
    printf OUTPUT "%-12s", "State_$lo_state{$lobyte}";
}
elsif (defined($hi_state{hibyte}))  {
    printf OUTPUT "%-12s", "State_$hi_state{hibyte}";
}
else {
    printf OUTPUT "%-7s", "ERROR!" ;
}
#-----

sub comps          #Components On-Line
{
    (@bytes) = unpack ("a"x4, $row[$n]);

    foreach $byte (@bytes)  {
        $val = pack "N", hex("$byte");
    }

    # Print out the bit values of $row, from low-order to high-order
    $hval = hex ("$row[$n]");

    $mask = hex("0001");
    for $i (0..10)  {
        $flag[$i] = ($hval & $mask);
        $hval = $hval >> 1;
        printf OUTPUT "%-3s", $flag[$i];
    }
}
#-----

```

\_\_END\_\_

## Appendix B – Summary and Reporting Program

```
Attribute VB_Name = "Module4"
Option Explicit

Sub main_macro()           'Just run this macro to create the whole monthly
report!
    Call arrange_columns   'Puts columns in proper order (Monarch screws them
up)
    Call summary_sheet4    'Creating & formatting 4 summary table sheets.
    Call summary_sheet3    'Order of creation is reversed so sheets will
appear in the
    Call summary_sheet2    'the workbook in the proper order, from left to
right.
    Call summary_sheet1
    Call summary_calcs    'Performs all of the calculations on the data and
places results in summary sheets.
    Call main_chart_macro  'Creates and formats all the summary charts.
    Call sheet_placement   'Organizes worksheet tabs in an appropriate
manner.
End Sub

Sub arrange_columns()       'Puts columns in proper order (Monarch screws them
up)
    Sheets("Raw_Data").Select
    Columns("CZ:CZ").Cut      'Date and Time
    Columns("A:A").Insert Shift:=xlToRight
    Sheets("Decimals").Select
    Columns("CF:CF").Cut      'Date and Time
    Columns("A:A").Insert Shift:=xlToRight
    Columns("CG:CG").Cut      'WTG1 Turb. Intensity
    Columns("AK:AK").Insert Shift:=xlToRight
    Columns("CH:CH").Cut      'WTG2 Turb. Intensity
    Columns("AR:AR").Insert Shift:=xlToRight
    Columns("CI:CI").Cut      'Wind Penetration
    Columns("AV:AV").Insert Shift:=xlToRight
    Range("AU2").Select
End Sub

Sub summary_sheet1()         'Creates a Summary Table sheet (Wind, Diesel, Load
Summary) for the
Attribute summary_sheet1.VB_Description = "Macro recorded 7/25/00 by jstack"
Attribute summary_sheet1.VB_ProcData.VB_Invoke_Func = "\n14"
    Sheets.Add                 'wind turbines, diesels, heating (dump) loads, and
village load
    ActiveSheet.name = "Wind, Diesel, Load Summary"
    'Column Headers
    ActiveCell.Range("B1").FormulaR1C1 = "Diesel 1"
    ActiveCell.Range("c1").FormulaR1C1 = "Diesel 2"
    ActiveCell.Range("d1").FormulaR1C1 = "Diesel 3"
    ActiveCell.Range("e1").FormulaR1C1 = "Total"
        Selection.Font.Italic = True
    ActiveCell.Range("B13").FormulaR1C1 = "Wind Turb 1"
    ActiveCell.Range("c13").FormulaR1C1 = "Wind Turb 2"
    ActiveCell.Range("d13").FormulaR1C1 = "Wind Turb 3"
    ActiveCell.Range("e13").FormulaR1C1 = "Total"
```

```

    Selection.Font.Italic = True
ActiveCell.Range("B31").FormulaR1C1 = "Local"
ActiveCell.Range("B32").FormulaR1C1 = "Heating Load"
ActiveCell.Range("c31").FormulaR1C1 = "Remote"
ActiveCell.Range("c32").FormulaR1C1 = "Heating Load"
ActiveCell.Range("b42").FormulaR1C1 = "Village Load"
'Row Headers
    ActiveCell.Range("a2").FormulaR1C1 = "Run Hours1"           'Diesels
    ActiveCell.Range("a2").Characters(start:=10,
Length:=1).Font.Superscript = True
    ActiveCell.Range("a3").FormulaR1C1 = "kWh Delivered"      'Diesels
    ActiveCell.Range("a4").FormulaR1C1 = "# Starts1"          'Diesels
    ActiveCell.Range("a4").Characters(start:=9, Length:=1).Font.Superscript
= True
    ActiveCell.Range("a14").FormulaR1C1 = "Run Hours1"         'Wind
Turb's
    ActiveCell.Range("a14").Characters(start:=10,
Length:=1).Font.Superscript = True
    ActiveCell.Range("a15").FormulaR1C1 = "kWh Delivered"      'Wind
Turb's
    ActiveCell.Range("a16").FormulaR1C1 = "#Contractor Closures1"   'Wind
Turb's
    ActiveCell.Range("a16").Characters(start:=21,
Length:=1).Font.Superscript = True
    ActiveCell.Range("a23").FormulaR1C1 = "Capacity Factor2,3,4"
'Wind Turb's
    ActiveCell.Range("a23").Characters(start:=16,
Length:=5).Font.Superscript = True
    ActiveCell.Range("a24").FormulaR1C1 = "Max. Inst. kW3"       'Wind
Turb's
    ActiveCell.Range("a24").Characters(start:=14,
Length:=1).Font.Superscript = True
    ActiveCell.Range("a26").FormulaR1C1 = "Avg. Hub Height Wind Speed4"
'Wind Turb's
    ActiveCell.Range("a26").Characters(start:=27,
Length:=1).Font.Superscript = True
    ActiveCell.Range("a27").FormulaR1C1 = "Max. Inst. Wind Speed"   'Wind
Turb's
    ActiveCell.Range("a28").FormulaR1C1 = "Avg. Turbulence Intensity4,5"
'Wind Turb's
    ActiveCell.Range("a28").Characters(start:=26,
Length:=3).Font.Superscript = True
    ActiveCell.Range("e26").FormulaR1C1 = " - - - "             'Wind Turb's
    ActiveCell.Range("e27").FormulaR1C1 = " - - - "             'Wind Turb's
    ActiveCell.Range("e28").FormulaR1C1 = " - - - "             'Wind Turb's
    ActiveCell.Range("a33").FormulaR1C1 = "kWh Consumed"        'Heating Loads
    ActiveCell.Range("a43").FormulaR1C1 = "Total kWh Consumed"
'Village Load
    ActiveCell.Range("a44").FormulaR1C1 = "Avg. kW1"            'Village
Load
    ActiveCell.Range("a44").Characters(start:=8,
Length:=1).Font.Superscript = True
    ActiveCell.Range("a45").FormulaR1C1 = "Max Inst. kW"        'Village Load
    ActiveCell.Range("a46").FormulaR1C1 = "Min Inst. kW"        'Village Load
    ActiveCell.Range("a6").FormulaR1C1 = "#Alarms"              'Alarms stuff
    ActiveCell.Range("a7").FormulaR1C1 = "Alarm Time"
    ActiveCell.Range("a8").FormulaR1C1 = "#Warnings"            'Warnings stuff

```

```

ActiveCell.Range("a9").FormulaR1C1 = "Warning Time"
Range("A6:A7").Font.ColorIndex = 3           'Making Alarms stuff red
Range("a8:a9").Font.ColorIndex = 10          'Making Warning stuff green
Range("A6:A9").Copy      'Copying the Alarms/Warnings stuff to the other
tables
Range("A18, a35").Select
ActiveSheet.Paste

'Formatting all the tables
Range("A1:A38, a43:a46").Select      'Formatting the Row headers
Selection.Font.name = "Times New Roman"
Selection.Font.Size = 9
Range("A2:E9, A14:E28, A33:C38, a43:b46").Select      'Borders around all the
tables
Selection.Borders(xlEdgeTop).Weight = xlThin
Selection.Borders(xlEdgeBottom).Weight = xlThin
Range("b1:d9, e1:e9, b13:d28, e13:e28, b31:c38, b42:b46").Select      'More
borders
Selection.Borders(xlEdgeRight).Weight = xlThin
Selection.Borders(xlEdgeLeft).Weight = xlThin
Range("B1:D1, B13:D13, B31:C32, b42").Select      'Dark gray background for
column headers
Selection.Interior.ColorIndex = 16
Range("E1, E13, A5:E5, A17:E17, A22:E22, A25:E25, A34:C34").Select
Selection.Interior.ColorIndex = 15                  'Light gray for table breaks
Cells.Select
Cells.Font.Bold = True
Cells.HorizontalAlignment = xlCenter
Cells.VerticalAlignment = xlBottom
Cells.EntireColumn.AutoFit
ActiveCell.Range("h2").FormulaR1C1 = "1. Run time & # of starts are
calculated from monthly time series data."
ActiveCell.Range("h3").FormulaR1C1 = "These values and actual values"
ActiveCell.Range("h4").FormulaR1C1 = "data (10 min)."
ActiveCell.Range("h5").FormulaR1C1 = "the actual accumulated values."
ActiveCell.Range("h14").FormulaR1C1 = "1. See note (#1) above"
ActiveCell.Range("h20").FormulaR1C1 = "2. Capacity factor values based on
turbine's peak power of 66kW"
ActiveCell.Range("h22").FormulaR1C1 = "3. Total capacity factor and Max
Inst. kW are based on WTG Total Avg kW"
ActiveCell.Range("h23").FormulaR1C1 = "Also, total capacity factor is based"
ActiveCell.Range("h24").FormulaR1C1 = "on sustained peak power output
of 66kW of 2 turbines."
ActiveCell.Range("h26").FormulaR1C1 = "4. All monthly averages based on #
of LOGGED hours, not total hrs in month"
ActiveCell.Range("h28").FormulaR1C1 = "5. Turb. Intensity = [(Max Wind
speed - Avg Wind speed) / Avg Wind speed]"
ActiveCell.Range("h44").FormulaR1C1 = "1. See note (#4) above"
Range("h2:h44").Select
Selection.Font.Bold = False
Selection.Font.Size = 7
Selection.HorizontalAlignment = xlLeft

```

```

    Range("b2:e3, b9:e9, b7:e7, b14:e15, b19:e19, b21:e28, b33:c33, b36:c36,
b38:c38, b43:b46").Select
        Selection.NumberFormat = "0.00"      'Chop off all the excess decimal
places
    Range("a1").Select
End Sub

Sub summary_sheet2()      'Creates and formats a 2nd Summary Table sheet
(Energy Storage, Control
    Sheets.Add      'Summary) for the Rotary Converter, ESS Shelter,
Battery Bank, and WDCP
    ActiveSheet.name = "Energy Storage, Control Summary"
'Column Headers
    ActiveCell.Range("B1").FormulaR1C1 = "AC"
    ActiveCell.Range("B2").FormulaR1C1 = "Machine"
    ActiveCell.Range("c1").FormulaR1C1 = "DC"
    ActiveCell.Range("c2").FormulaR1C1 = "Machine"
    ActiveCell.Range("d1").FormulaR1C1 = "Rotary"
    ActiveCell.Range("d2").FormulaR1C1 = "Converter"
    ActiveCell.Range("b16").FormulaR1C1 = "Battery"
    ActiveCell.Range("b17").FormulaR1C1 = "Bank"
    ActiveCell.Range("b50").FormulaR1C1 = "ESS"
    ActiveCell.Range("b51").FormulaR1C1 = "Shelter"
    ActiveCell.Range("b68").FormulaR1C1 = "Wind Diesel"
    ActiveCell.Range("b69").FormulaR1C1 = "Control Panel"
    ActiveCell.Range("b70").FormulaR1C1 = "(WDCP)"
'Row Headers
    ActiveCell.Range("a3").FormulaR1C1 = "Run Hours1"      'Rotary Converter
stuff
    ActiveCell.Range("a3").Characters(start:=10, Length:=1).Font.Superscript
= True
    ActiveCell.Range("a4").FormulaR1C1 = "RC Spin Losses (kWh)"
    ActiveCell.Range("a5").FormulaR1C1 = "#Contractor Closures1"
    ActiveCell.Range("a5").Characters(start:=21, Length:=1).Font.Superscript
= True
    ActiveCell.Range("b4").FormulaR1C1 = "'--- -"
    ActiveCell.Range("b9").FormulaR1C1 = "'--- -"
    ActiveCell.Range("b10").FormulaR1C1 = "'--- -"
    ActiveCell.Range("c4").FormulaR1C1 = "'--- -"
    ActiveCell.Range("d3").FormulaR1C1 = "'--- -"
    ActiveCell.Range("d5").FormulaR1C1 = "'--- -"
    ActiveCell.Range("a23").FormulaR1C1 = "Beginning SOC"      'Battery Bank
stuff
    ActiveCell.Range("a24").FormulaR1C1 = "Ending SOC"
    ActiveCell.Range("a26").FormulaR1C1 = "Charge kWh"
    ActiveCell.Range("a27").FormulaR1C1 = "Discharge kWh"
    ActiveCell.Range("a28").FormulaR1C1 = "Energy Losses (kWh)1"
    ActiveCell.Range("a28").Characters(start:=20,
Length:=1).Font.Superscript = True
    ActiveCell.Range("a29").FormulaR1C1 = "Energy Efficiency2"
    ActiveCell.Range("a29").Characters(start:=18,
Length:=1).Font.Superscript = True
    ActiveCell.Range("a31").FormulaR1C1 = "Charge Ah"
    ActiveCell.Range("a32").FormulaR1C1 = "Discharge Ah"
    ActiveCell.Range("a33").FormulaR1C1 = "Charge Losses (Ah)1"
    ActiveCell.Range("a33").Characters(start:=19,
Length:=1).Font.Superscript = True

```

```

    ActiveCell.Range("a34").FormulaR1C1 = "Charge Efficiency3"
    ActiveCell.Range("a34").Characters(start:=18,
Length:=1).Font.Superscript = True
    ActiveCell.Range("a36").FormulaR1C1 = "Avg. Voltage4"
    ActiveCell.Range("a36").Characters(start:=13,
Length:=1).Font.Superscript = True
    ActiveCell.Range("a37").FormulaR1C1 = "Max Inst. Voltage"
    ActiveCell.Range("a38").FormulaR1C1 = "Min Inst. Voltage"
    ActiveCell.Range("a40").FormulaR1C1 = "Avg. Temp4"
    ActiveCell.Range("a40").Characters(start:=10,
Length:=1).Font.Superscript = True
    ActiveCell.Range("a41").FormulaR1C1 = "Max Temp5"
    ActiveCell.Range("a41").Characters(start:=9,
Length:=1).Font.Superscript = True
    ActiveCell.Range("a42").FormulaR1C1 = "Min Temp5"
    ActiveCell.Range("a42").Characters(start:=9,
Length:=1).Font.Superscript = True
    ActiveCell.Range("a57").FormulaR1C1 = "Avg. Indoor Temp1" 'ESS Shelter
stuff
    ActiveCell.Range("a57").Characters(start:=17,
Length:=1).Font.Superscript = True
    ActiveCell.Range("a58").FormulaR1C1 = "Max Indoor Temp2"
    ActiveCell.Range("a58").Characters(start:=16,
Length:=1).Font.Superscript = True
    ActiveCell.Range("a59").FormulaR1C1 = "Min Indoor Temp2"
    ActiveCell.Range("a59").Characters(start:=16,
Length:=1).Font.Superscript = True
    ActiveCell.Range("a61").FormulaR1C1 = "Avg. Outdoor Temp1"
    ActiveCell.Range("a61").Characters(start:=18,
Length:=1).Font.Superscript = True
    ActiveCell.Range("a62").FormulaR1C1 = "Max Outdoor Temp2"
    ActiveCell.Range("a62").Characters(start:=17,
Length:=1).Font.Superscript = True
    ActiveCell.Range("a63").FormulaR1C1 = "Min Outdoor Temp2"
    ActiveCell.Range("a63").Characters(start:=17,
Length:=1).Font.Superscript = True
    ActiveCell.Range("a7").FormulaR1C1 = "#Alarms"           'Alarms (all
tables)
    ActiveCell.Range("a8").FormulaR1C1 = "Alarm Time"
    ActiveCell.Range("a9").FormulaR1C1 = "#Warnings"        'Warnings (all
tables)
    ActiveCell.Range("a10").FormulaR1C1 = "Warning Time"
Range("A7:A8").Select
Selection.Font.ColorIndex = 3      'Making Alarms stuff red
Range("a9:a10").Select
Selection.Font.ColorIndex = 10     'Making Warning stuff green
Range("A7:A10").Copy
Range("a71, a52, a18").Select      'Copying Alarms/Warnings stuff to the
other tables
ActiveSheet.Paste

'Formatting all the tables--Fonts and Backgrounds
Range("B1:D2, b16:B17, b50:b51, b68:b70").Select
Selection.Interior.ColorIndex = 16      'Dark gray background for column
headers
Range("A22:B22, A25:B25, A30:B30, A35:B35, A39:B39, a60:b60, a56:b56,
A6:D6").Select

```

```

    Selection.Interior.ColorIndex = 15      'Light Gray background for breaks
in tables
    Range("A3:d10, A18:b42, a52:b63, a71:b74").Select      'Borders around tables
        Selection.Borders(xlEdgeTop).Weight = xlThin
        Selection.Borders(xlEdgeBottom).Weight = xlThin
    Range("B1:D10, b16:b42, b50:b63, b68:b74").Select      'more borders around
tables
        Selection.Borders(xlEdgeRight).Weight = xlThin
        Selection.Borders(xlEdgeLeft).Weight = xlThin
    Range("A:A").Select                      'Formatting Row Headers
        Selection.Font.name = "Times New Roman"
        Selection.Font.Size = 9
Cells.Select
    Cells.HorizontalAlignment = xlCenter
    Cells.VerticalAlignment = xlBottom
    Cells.Font.Bold = True
    Cells.EntireColumn.AutoFit
    ActiveCell.Range("g3").FormulaR1C1 = "1. Run time & # of starts are
calculated from monthly time series data."
    ActiveCell.Range("g4").FormulaR1C1 = "           There may be discrepancies b/w
these values and actual values"
    ActiveCell.Range("g5").FormulaR1C1 = "           due to resolution of time series
data (10 min)."
    ActiveCell.Range("g6").FormulaR1C1 = "           In the future, we plan to send
the actual accumulated values."
    ActiveCell.Range("g27").FormulaR1C1 = "1. Losses are completely accurate
only if beginning SOC is equal to"
    ActiveCell.Range("g28").FormulaR1C1 = "           ending SOC. Also, 'Losses' could
be negative if ending SOC is less."
    ActiveCell.Range("g30").FormulaR1C1 = "2. Energy Efficiency = Discharge
kWh / Charge kWh"
    ActiveCell.Range("g34").FormulaR1C1 = "3. Charge Efficiency = Discharge Ah
/ Charge Ah"
    ActiveCell.Range("g36").FormulaR1C1 = "4. All monthly averages based on #
of LOGGED hours, not total hrs in month."
    ActiveCell.Range("g41").FormulaR1C1 = "5. Max & Min Temps are based on 10
min averages, not instantaneous values."
    ActiveCell.Range("g57").FormulaR1C1 = "1. All monthly averages based on #
of LOGGED hours, not total hrs in month."
    ActiveCell.Range("g59").FormulaR1C1 = "2. Max & Min Temps are based on 10
min averages, not instantaneous values."

Range("g3:g59").Select
    Selection.Font.Bold = False
    Selection.Font.Size = 7
    Selection.HorizontalAlignment = xlLeft

Range("b3:d4, b10:d10, b8:d8, b16, b18, b20:b39, b57:b63, b72, b74, b53,
b55").Select
    Selection.NumberFormat = "0.00"          'Chopping off excess decimals
Range("a1").Select
End Sub

Sub summary_sheet3()          'Creates and formats a 3rd Summary Table sheet
(System Parameters Summary)
    Sheets.Add                  'for the AC Bus stuff and info on the System State
& Mode

```

```

ActiveSheet.name = "System Parameters Summary"
'Column Headers
ActiveCell.Range("B1").FormulaR1C1 = "System"
ActiveCell.Range("c1").FormulaR1C1 = "Parameters"
ActiveCell.Range("a3").FormulaR1C1 = "# AC Bus Alarms:"
ActiveCell.Range("a4").FormulaR1C1 = "Alarm Time (hrs):"
ActiveCell.Range("d3").FormulaR1C1 = "# AC Bus Warnings:"
ActiveCell.Range("d4").FormulaR1C1 = "Warning Time (hrs):"
ActiveCell.Range("a7").FormulaR1C1 = "Total Hours of:"
ActiveCell.Range("d6").FormulaR1C1 = "kWh Delivered"
ActiveCell.Range("d7").FormulaR1C1 = "(Wind-to-Load) with:"

'Row Headers
ActiveCell.Range("a8").FormulaR1C1 = "Diesel ON:"
ActiveCell.Range("a9").FormulaR1C1 = "Diesel OFF:"
ActiveCell.Range("a11").FormulaR1C1 = "State 0"
ActiveCell.Range("a12").FormulaR1C1 = "State 1A"
ActiveCell.Range("a13").FormulaR1C1 = "State 1B"
ActiveCell.Range("a14").FormulaR1C1 = "State 2A"
ActiveCell.Range("a15").FormulaR1C1 = "State 2B"
ActiveCell.Range("a16").FormulaR1C1 = "State 3"
ActiveCell.Range("a17").FormulaR1C1 = "State 4"
ActiveCell.Range("a18").FormulaR1C1 = "State 5"
ActiveCell.Range("a19").FormulaR1C1 = "State 6"
ActiveCell.Range("a20").FormulaR1C1 = "State 7"
ActiveCell.Range("d8").FormulaR1C1 = "Diesel ON1:"
ActiveCell.Range("d8").Characters(start:=10,
Length:=1).Font.Superscript = True
ActiveCell.Range("d9").FormulaR1C1 = "Diesel OFF2:"
ActiveCell.Range("d9").Characters(start:=11,
Length:=1).Font.Superscript = True
ActiveCell.Range("d10").FormulaR1C1 = "Total3:"
ActiveCell.Range("d10").Characters(start:=6,
Length:=1).Font.Superscript = True
ActiveCell.Range("a22").FormulaR1C1 = "Manual Mode6"
ActiveCell.Range("a22").Characters(start:=12,
Length:=1).Font.Superscript = True
ActiveCell.Range("a23").FormulaR1C1 = "Mode 0"
ActiveCell.Range("a24").FormulaR1C1 = "Mode 1"
ActiveCell.Range("a25").FormulaR1C1 = "Mode 2"
ActiveCell.Range("a26").FormulaR1C1 = "Mode 3"
ActiveCell.Range("d14").FormulaR1C1 = "Total Wind"
ActiveCell.Range("d15").FormulaR1C1 = "Penetration4:"
ActiveCell.Range("d15").Characters(start:=12, Length:=1).Font.Superscript
= True
ActiveCell.Range("d17").FormulaR1C1 = "Max Penetration5:"
ActiveCell.Range("d17").Characters(start:=16, Length:=1).Font.Superscript
= True

'Formatting the Row & Column Headers
Range("b1:c1").Select      'Formatting the Sheet Title
Selection.Font.Size = 12
Selection.Borders(xlEdgeBottom).Weight = xlMedium
Selection.Interior.ColorIndex = 15
Range("A7, d14:d15, d17, d6:e7, B1:C1").Select    'Bold-facing the column
headers
Selection.Font.Bold = True
Range("d10").Select

```

```

Selection.Font.Italic = True
Selection.Font.Size = 11
Range("A3:A4").Select
    Selection.Font.ColorIndex = 3      'Making Alarms stuff red
Range("D3:D4").Select
    Selection.Font.ColorIndex = 10     'Making Warning stuff green
Range("a10:b10, a21:b21").Select
    Selection.Interior.ColorIndex = 15   'Light gray background for table
breaks
Range("a7:b7, d7:e7").Select
    Selection.Borders(xlEdgeBottom).Weight = xlThin
Range("a1").Select
ActiveCell.Range("g7").FormulaR1C1 = "1. Wind kWh delivered to Village
load during Diesel ON operation is the amount of the"
ActiveCell.Range("g8").FormulaR1C1 = "village load supplied by wind
energy during times when at least one diesel was on-line."
ActiveCell.Range("g9").FormulaR1C1 = "2. Wind kWh delivered to Village
load during Diesel OFF operation is the amount of the"
ActiveCell.Range("g10").FormulaR1C1 = "village load supplied by wind
energy during times when all diesels were shut off."
ActiveCell.Range("g11").FormulaR1C1 = "3. Total Wind kWh delivered to
Village load is the total amount of the village load supplied by wind
energy."
ActiveCell.Range("g15").FormulaR1C1 = "4. Wind Energy Penetration = Total
Wind Turbine kWh / Total Village Load kWh"
ActiveCell.Range("g17").FormulaR1C1 = "5. 10min. Wind Energy Penetration =
Avg. Total Wind Power / Avg. Village Load over a 10min. period."
ActiveCell.Range("g22").FormulaR1C1 = "6. In Manual Mode, diesels are
started & stopped by the operator, HOWEVER, they may be started and stopped
from EITHER the"
ActiveCell.Range("g23").FormulaR1C1 = "WDCP Touchscreen or the AVEC
Diesel Control Panel, depending on the position of the Manual/Auto switch"
ActiveCell.Range("g24").FormulaR1C1 = "(on the AVEC Diesel Control
Panel). In Manual, diesels are controlled by the AVEC DCP. In Auto, diesels
are controlled"
ActiveCell.Range("g25").FormulaR1C1 = "from the WDCP Touchscreen. We
aren't currently sending info to allow us to differentiate b/w these 2 sub-
modes of operation."
Range("g7:g25").Select
    Selection.Font.Bold = False
    Selection.Font.Size = 7
    Selection.Horizontal_ALIGNMENT = xlLeft

Cells.Select
Cells.EntireColumn.AutoFit      'Fitting data to cells
Range("b4:e26").Select
    Selection.NumberFormat = "0.00"      'Chopping off excess decimals
ActiveSheet.PageSetup.PrintGridlines = True
Range("a1").Select
End Sub

Sub summary_sheet4()      'Creates and formats a 4th summary sheet (Data
Summary) with info on
    Sheets.Add      'the total amount of data for the month and any data
that's missing
    ActiveSheet.name = "Data Summary"
    'Table Titles

```

```

ActiveCell.Range("a1").FormulaR1C1 = "Data"
ActiveCell.Range("b1").FormulaR1C1 = "Acquisition System"
ActiveCell.Range("a7").FormulaR1C1 = "Missing"
ActiveCell.Range("b7").FormulaR1C1 = "Data"
ActiveCell.Range("c7").FormulaR1C1 = "Segments"
'Row Headers for "Data Acquisition System" Table
ActiveCell.Range("a2").FormulaR1C1 = "Hours in Month:"
ActiveCell.Range("a3").FormulaR1C1 = "Logged Hours:"
ActiveCell.Range("a4").FormulaR1C1 = "Missing Hours1:"
    ActiveCell.Range("a4").Characters(start:=14,
Length:=1).Font.Superscript = True
'Column headers for "Missing Data Segments" Table
ActiveCell.Range("a8").FormulaR1C1 = "Segment #"
ActiveCell.Range("b8").FormulaR1C1 = "Beginning of Segment (Date & Time)"
ActiveCell.Range("c8").FormulaR1C1 = "End of Segment (Date & Time)"

'Formatting stuff
Range("a1:b1, a7:c7").Select      'Formatting Table titles
With Selection.Font
    .Bold = True
    .Name = "MS Sans Serif"
    .Size = 12
    .ColorIndex = xlAutomatic
End With
Selection.Interior.ColorIndex = 15      'Light gray background for table
titles
Selection.Borders(xlEdgeBottom).Weight = xlMedium      'Underlining table
titles
Range("a2:a4, a8:c8").Select
    Selection.Font.Bold = True      'Bold-facing column and row headers
Range("a1").Select
ActiveCell.Range("e4").FormulaR1C1 = "1. Missing data represents times
when the WDCP PLC was off"
ActiveCell.Range("e5").FormulaR1C1 = " or not in run mode, or when the
data logging system failed."
Range("e4:e5").Select
    Selection.Font.Bold = False
    Selection.Font.Size = 7
    Selection.HorizontalAlignment = xlLeft
Cells.Select
    Selection.HorizontalAlignment = xlCenter      'Centering and fitting all
cells
    Selection.EntireColumn.AutoFit
Range("b2:b4").Select
    Selection.NumberFormat = "0.00"      'Chopping off excess decimals
Range("A1").Select
End Sub

Sub summary_calcs()
    Call runhrs_starts      'Run hours and Starts for Diesels and Turbines
    Call kWh                 'kWh delivered/consumed by diesels, turbines,
village load, dump loads
    Call alarms_warnings     'Alarms/Warnings (#'s and total time) from
"Alarms" sheet
    Call alarms_warnings2   'Alarms/Warnings (#'s and total time) from
"ESS_Alarms" sheet
    Call wt_capfactor       'Capacity factor for 1,2 & total

```

```

    Call max_min_avg      'wind speeds, turbulence, voltage, temps, lots of
stuff
    Call SOC             'Battery bank's beginning & ending SOC's
    Call charge_discharge 'Battery Bank Charge, Discharge (kWh & Ah),
Losses, and Efficiencies
    Call state_mode      'Amount of time in each State & Mode (System
Parameters Summary sheet)
    Call don_dOFF         'Amount of time in & kWh delivered (wind-to-load)
in diesel ON & OFF modes
    Call data_acq          'Data Summary sheet stuff (hrs in month, etc.)
    Call gaps              'Fills in the missing data segment gaps in
'Raw_Data' sheet
    Call missing_segments   'counts up # missing data segments, prints their
start and end times to 'Data Summary' sheet
End Sub

Sub runhrs_starts()
    Dim i As Integer, j As Integer, LastRow As Integer
    Dim sum As Integer, startsum As Integer
    Dim runhrs(4 To 13) As Double, count(4 To 13) As Double
    Dim valu As Integer, prev As Integer
    Sheets("Raw_Data").Select
    LastRow = Application.CountA(ActiveSheet.Range("A:A"))           'Get last used
row of data sheet
    For i = 4 To 13          'Columns D to I     (run hours for diesels and wind
turb)
        count(i) = 0
        runhrs(i) = 0
        startsum = 0
        sum = 0
        For j = 3 To LastRow      'sum up all rows in each column
            valu = val(ActiveSheet.Columns(i).Cells(j).Value)
            prev = val(ActiveSheet.Columns(i).Cells(j - 1).Value)
            If valu = 1 And prev = 0 Then
                startsum = startsum + 1    'Count as a start if it goes from 0 to
1 (OFF to ON)
            End If
            sum = sum + valu        'Add up # of ON time periods
        Next j
        count(i) = startsum      'Total # of starts
        sum = sum + val(ActiveSheet.Columns(i).Cells(2).Value)    'Account for
skipped 1st row (j=2)
        runhrs(i) = sum / 6      'Convert # time periods (10 min. each) to hours
    Next i
    Sheets("Wind, Diesel, Load Summary").Select
    ActiveSheet.Range("b2").Value = runhrs(4)           'Diesel Run Hours
    ActiveSheet.Range("c2").Value = runhrs(5)
    ActiveSheet.Range("d2").Value = runhrs(6)
    ActiveSheet.Range("b14").Value = runhrs(7)          'Wind Turbine Run Hours
    ActiveSheet.Range("c14").Value = runhrs(8)
    ActiveSheet.Range("d14").Value = runhrs(9)
    ActiveSheet.Range("E2").FormulaR1C1 = "=SUM(RC[-3]:RC[-1])" 'Total
Diesel run hours
    ActiveSheet.Range("E14").FormulaR1C1 = "=SUM(RC[-3]:RC[-1])" 'Total
Wind Turbine run hours
    ActiveSheet.Range("b4").Value = count(4)           'Diesel starts
    ActiveSheet.Range("c4").Value = count(5)

```

```

ActiveSheet.Range("d4").Value = count(6)
ActiveSheet.Range("b16").Value = count(7)      'wind turbine contractor
closures
    ActiveSheet.Range("c16").Value = count(8)
    ActiveSheet.Range("d16").Value = count(9)
    ActiveSheet.Range("E4").FormulaR1C1 = "=SUM(RC[-3]:RC[-1])"      'Total
Diesel starts
    ActiveSheet.Range("E16").FormulaR1C1 = "=SUM(RC[-3]:RC[-1])"      'Total
wind turbine contractor closures

Sheets("Energy Storage, Control Summary").Select
    ActiveSheet.Range("b3").Value = runhrs(12)      'AC Machine Run Hours
    ActiveSheet.Range("c3").Value = runhrs(13)      'DC Machine Run Hours
    ActiveSheet.Range("b5").Value = count(12)      'AC Machine starts
    ActiveSheet.Range("c5").Value = count(13)      'DC Machine starts
End Sub

Sub kWh()      'kWh delivered by diesels, turbines; kWh consumed by village
load, heating loads
    Dim i As Integer, j As Integer, LastRow As Integer
    Dim sum As Integer, kWhrs(21 To 57) As Double, valu As Integer
    Sheets("Decimals").Select
    LastRow = Application.CountA(ActiveSheet.Range("A:A"))      'Get last used
row of data sheet
    For i = 21 To 57      'Columns U-BE (kW Avg for diesels, wind turbs, village
load, and dump loads)
        kWhrs(i) = 0
        sum = 0
        For j = 2 To LastRow      'sum up all rows in each column
            valu = val(ActiveSheet.Columns(i).Cells(j).Value)
            sum = sum + valu
        Next j
        kWhrs(i) = sum / 6      'Convert kW to kWh
    Next i
    Sheets("Wind, Diesel, Load Summary").Select
        ActiveSheet.Columns("b").Cells(3).Value = kWhrs(21)      'Diesel kWh
delivered (D1)
        ActiveSheet.Columns("c").Cells(3).Value = kWhrs(22)      '(D2)
        ActiveSheet.Columns("d").Cells(3).Value = kWhrs(23)      '(D3)
        ActiveSheet.Columns("e").Cells(3).Value = "=SUM(RC[-3]:RC[-1])"
'Total Diesel kWh
        ActiveSheet.Columns("b").Cells(15).Value = kWhrs(31)      'Wind Turb kWh
delivered (WT1)
        ActiveSheet.Columns("c").Cells(15).Value = kWhrs(38)      '(WT2)
        ActiveSheet.Columns("e").Cells(15).Value = "=SUM(RC[-3]:RC[-1])"
'Total WT kWh
        ActiveSheet.Columns("b").Cells(43).Value = kWhrs(57)      'Village Load kWh
consumed
        ActiveSheet.Columns("b").Cells(33).Value = kWhrs(49)      'Local DL kWh
consumed
        ActiveSheet.Columns("c").Cells(33).Value = kWhrs(50)      'Remote DL kWh
consumed
    Sheets("System Parameters Summary").Select
        ActiveSheet.Columns("e").Cells(15).Value = (kWhrs(45) / kWhrs(57)) 'Wind
Energy Penetration (Total WTG kWh/Total Village kWh)
End Sub

```

```

Sub alarms_warnings()      '# of Alarms and Warnings as well as total Alarm
and Warning time
    'for Diesels, Wind Turbines, WDCP, and AC Bus. All values
taken from 'Alarms' sheet.
    Dim i As Integer, j As Integer, val As Integer, prev As Integer
    Dim LastRow As Integer, count(118 To 152) As Double, starts As Integer
    Dim time As Integer, hours(118 To 152) As Double
    Sheets("Alarms").Select
    LastRow = Application.CountA(ActiveSheet.Range("A:A"))      'Get last used
row of data sheet
    For i = 118 To 152          'Columns DN to EV   (general alarms, warnings,
cautions--16 categories)
        count(i) = 0
        starts = 0
        hours(i) = 0
        time = 0
        For j = 3 To LastRow      'sum up all rows in each column
            val = val(ActiveSheet.Columns(i).Cells(j).Value)
            prev = val(ActiveSheet.Columns(i).Cells(j - 1).Value)
            If val = 1 And prev = 0 Then      'Count as an Alarm/Warning
occurrence if it goes
                starts = starts + 1          'from 0 to 1.
            End If
            time = time + val           'Get total Alarm time (val is always 0
or 1, so you don't
        Next j                      'need an If statement).
        count(i) = starts           'count = total # of unique alarm occurrences
        hours(i) = time / 6         'Convert Alarm/Warning time to hours
    Next i
    Sheets("Wind, Diesel, Load Summary").Select
    ActiveSheet.Range("b6").Value = count(124)      'D1 alarms
    ActiveSheet.Range("b7").Value = hours(124)        'D1 alarm time
    ActiveSheet.Range("c6").Value = count(132)        'D2 alarms
    ActiveSheet.Range("c7").Value = hours(132)        'D2 alarm time
    ActiveSheet.Range("d6").Value = count(129)        'D3 alarms
    ActiveSheet.Range("d7").Value = hours(129)        'D3 alarm time
    ActiveSheet.Range("b8").Value = count(125)        'D1 warnings
    ActiveSheet.Range("b9").Value = hours(125)        'D1 warning time
    ActiveSheet.Range("c8").Value = count(128)        'D2 warnings
    ActiveSheet.Range("c9").Value = hours(128)        'D2 warning time
    ActiveSheet.Range("d8").Value = count(130)        'D3 warnings
    ActiveSheet.Range("d9").Value = hours(130)        'D3 warning time
    ActiveSheet.Range("b18").Value = count(145)        'WT1 alarms
    ActiveSheet.Range("b19").Value = hours(145)        'WT1 alarm time
    ActiveSheet.Range("c18").Value = count(148)        'WT2 alarms
    ActiveSheet.Range("c19").Value = hours(148)        'WT2 alarm time
    ActiveSheet.Range("d18").Value = count(151)        'WT3 alarms
    ActiveSheet.Range("d19").Value = hours(151)        'WT3 alarm time
    ActiveSheet.Range("b20").Value = count(146)        'WT1 warnings
    ActiveSheet.Range("b21").Value = hours(146)        'WT1 warning time
    ActiveSheet.Range("c20").Value = count(149)        'WT2 warnings ds
    ActiveSheet.Range("c21").Value = hours(149)        'WT2 warning time
    ActiveSheet.Range("d20").Value = count(152)        'WT3 warnings
    ActiveSheet.Range("d21").Value = hours(152)        'WT3 warning time
    ActiveSheet.Range("e6").FormulaR1C1 = "=SUM(RC[-3]:RC[-1])"      'Total
Diesel alarms

```

```

        ActiveSheet.Range("e7").FormulaR1C1 = "=SUM(RC[-3]:RC[-1])"      'Total
Diesel time
        ActiveSheet.Range("e8").FormulaR1C1 = "=SUM(RC[-3]:RC[-1])"      'Total
Diesel warnings
        ActiveSheet.Range("e9").FormulaR1C1 = "=SUM(RC[-3]:RC[-1])"      'Total
Diesel time
        ActiveSheet.Range("e18").FormulaR1C1 = "=SUM(RC[-3]:RC[-1])"      'Total
Wind turb alarms
        ActiveSheet.Range("e19").FormulaR1C1 = "=SUM(RC[-3]:RC[-1])"      'Total
Wind turb alarm time
        ActiveSheet.Range("e20").FormulaR1C1 = "=SUM(RC[-3]:RC[-1])"      'Total
Wind turbine warnings
        ActiveSheet.Range("e21").FormulaR1C1 = "=SUM(RC[-3]:RC[-1])"      'Total
Wind turbine warning time

Sheets("Energy Storage, Control Summary").Select
        ActiveSheet.Range("b71").Value = count(121)                      'WDCP alarms
        ActiveSheet.Range("b72").Value = hours(121)                        'WDCP alarm time
        ActiveSheet.Range("b73").Value = count(123)                        'WDCP warnings
        ActiveSheet.Range("b74").Value = hours(123)                        'WDCP warning time
Sheets("System Parameters Summary").Select
        ActiveSheet.Range("b3").Value = count(118)                        'AC Bus alarms
        ActiveSheet.Range("b4").Value = hours(118)                         'AC Bus alarm time
        ActiveSheet.Range("e3").Value = count(119)                        'AC Bus warnings
        ActiveSheet.Range("e4").Value = hours(119)                         'AC Bus warning time
End Sub

Sub alarms_warnings2()          '# of Alarms and Warnings as well as total Alarm
and Warning time
                                         'for Dump Loads, ESS Shelter, Battery Bank, AC
machine, DC machine,
                                         'and Rotary Converter. All values taken from 'ESS
Alarms' sheet.
    Dim i As Integer, j As Integer, valu As Integer, prev As Integer
    Dim LastRow As Integer, count(93 To 109) As Double, starts As Integer
    Dim time As Integer, hours(93 To 109) As Double
    Sheets("ESS_Alarms").Select
        LastRow = Application.CountA(ActiveSheet.Range("A:A"))           'Get last used
row of data sheet
        For i = 93 To 109   'Columns CO-DE (general alarms, warnings, cautions
columns--16 categories)
            count(i) = 0
            starts = 0
            hours(i) = 0
            time = 0
            For j = 3 To LastRow           'sum up all rows in each column
                valu = val(ActiveSheet.Columns(i).Cells(j).Value)
                prev = val(ActiveSheet.Columns(i).Cells(j - 1).Value)
                If valu = 1 And prev = 0 Then
                    starts = starts + 1
                End If
                time = time + valu
            Next j
            count(i) = starts
            hours(i) = time / 6          'Convert Alarm/Warning time to hours
        Next i
    Sheets("Wind, Diesel, Load Summary").Select

```

```

ActiveSheet.Range("b35").Value = count(96)      'Local DL alarms
ActiveSheet.Range("b36").Value = hours(96)        'Local DL alarm time
ActiveSheet.Range("b37").Value = count(97)        'Local DL warnings
ActiveSheet.Range("b38").Value = hours(97)        'Local DL warning time
ActiveSheet.Range("c35").Value = count(100)       'Remote DL alarms
ActiveSheet.Range("c36").Value = hours(100)       'Remote DL alarm time
ActiveSheet.Range("c37").Value = count(98)        'Remote warnings
ActiveSheet.Range("c38").Value = hours(98)        'Remote warning time

Sheets("Energy Storage, Control Summary").Select
    ActiveSheet.Range("b7").Value = count(101)      'AC machine alarms NOTE:
No AC machine warnings
    ActiveSheet.Range("b8").Value = hours(101)      'AC machine alarm time
    ActiveSheet.Range("c7").Value = count(102)      'DC machine alarms
    ActiveSheet.Range("c8").Value = hours(102)      'DC machine alarm time
    ActiveSheet.Range("c9").Value = count(104)      'DC machine warnings
    ActiveSheet.Range("c10").Value = hours(104)     'DC machine warning time
    ActiveSheet.Range("d7").Value = count(105)      'Rotary converter alarms
    ActiveSheet.Range("d8").Value = hours(105)      'Rotary converter alarm
time
    ActiveSheet.Range("d9").Value = count(106)      'Rotary converter warnings
    ActiveSheet.Range("d10").Value = hours(106)     'Rotary converter warning
ActiveSheet.Range("b18").Value = count(108)      'Battery bank alarms
ActiveSheet.Range("b19").Value = hours(108)      'Battery bank alarm time
ActiveSheet.Range("b20").Value = count(109)      'Battery bank warnings
ActiveSheet.Range("b21").Value = hours(109)      'Battery bank warning time
ActiveSheet.Range("b52").Value = count(93)        'ESS Shelter alarms
ActiveSheet.Range("b53").Value = hours(93)        'ESS Shelter alarm time
ActiveSheet.Range("b54").Value = count(94)        'ESS Shelter warnings
ActiveSheet.Range("b55").Value = hours(94)        'ESS Shelter warning time
End Sub

Sub wt_capfactor()           'Monthly avg. kW / Peak Power Rate (66kW)
    Dim i As Integer, j As Integer, LastRow As Integer
    Dim sum As Integer, kW(31 To 45) As Double, valu As Integer
    Dim cap(31 To 45) As Double, n As Integer, peak As Integer
Sheets("Decimals").Select
    LastRow = Application.CountA(ActiveSheet.Range("A:A"))      'Get last used
row of data sheet
    sum = 0
    n = 2          '# of currently operating turbines (for Total Cap. Factor
calculation)
    peak = 66      'Peak power rating of turbines = 66kW
    For i = 31 To 45 Step 7        'Columns U to X (kW Avg for turbines 1,2 &
total)
        kW(i) = 0
        sum = 0
        For j = 2 To LastRow      'sum up all rows in each column
            valu = val(ActiveSheet.Columns(i).Cells(j).Value)
            sum = sum + valu
        Next j
        kW(i) = sum / (LastRow - 1)      'Monthly average kW
        cap(i) = kW(i) / peak          'Capacity factor for individual
turbines
        cap(45) = kW(45) / (n * peak)    'Total Cap.Factor (Total Mo'ly avg.
kW/total peak power)

```

```

    Next i
Sheets("Wind, Diesel, Load Summary").Select
    ActiveSheet.Range("b23").Value = cap(31)      'WT1 capacity
    ActiveSheet.Range("c23").Value = cap(38)      'WT2 capacity
    ActiveSheet.Range("e23").Value = cap(45)      'Total WT capacity
End Sub

Sub max_min_avg()      'Max. Instantaneous kW and hub height wind speed for
turbines
    'Average wind speed, turbulence intensity      'Avg, max, min
Village Load
    'Avg,Max&Min Battery Bank Voltage & Temp, ESS Shelter Indoor
& Outdoor temps
    'Also Rotary Converter Spinning Losses (kWh)
Dim i As Integer, j As Integer, LastRow As Integer
Dim max(32 To 87) As Double, min(32 To 87) As Double
Dim valu As Double, prev As Double
Dim sum As Double, avg(32 To 87) As Double, kWh(32 To 87) As Double
Sheets("Decimals").Select
    LastRow = Application.CountA(ActiveSheet.Range("A:A"))      'Get last used
row of data sheet
    For i = 32 To 87      'Columns AF to CI
        max(i) = val(ActiveSheet.Columns(i).Cells(2).Value)      'Initialize max
and min values
        min(i) = val(ActiveSheet.Columns(i).Cells(2).Value)
        avg(i) = 0
        sum = 0
        kWh(i) = 0      'For R.C. Spinning Losses
        For j = 2 To LastRow
            valu = val(ActiveSheet.Columns(i).Cells(j).Value)
            sum = sum + valu      'running total of all values in the
column
            If valu > max(i) Then      'Compare each value to current max value
                max(i) = valu
            ElseIf valu < min(i) Then      'Compare each value to current min
value
                min(i) = valu
            End If
        Next j
        avg(i) = (sum / (LastRow - 1))
        kWh(i) = sum / 6      'Convert kW to kWh
    Next i
Sheets("Wind, Diesel, Load Summary").Select
    ActiveSheet.Range("b24").Value = max(32)      'Wind Turb Max Instantaneous
kW (WT1)
    ActiveSheet.Range("c24").Value = max(39)      '(WT2)
    ActiveSheet.Range("e24").Value = max(46)      '(Total)
    ActiveSheet.Range("b27").Value = max(36)      'Wind Turb Max Instantaneous
wind speed (WT1)
    ActiveSheet.Range("c27").Value = max(43)      '(WT2)
    ActiveSheet.Range("b26").Value = avg(35)      'Wind Turb average wind speed
(WT1)
    ActiveSheet.Range("c26").Value = avg(42)      '(WT2)
    ActiveSheet.Range("b28").Value = avg(37)      'Wind Turb avg turbulence
intensity (WT1)
    ActiveSheet.Range("c28").Value = avg(44)      '(WT2)
    ActiveSheet.Range("b44").Value = avg(57)      'Village load kW (avg)

```

```

ActiveSheet.Range("b45").Value = max(58)      '(Max)
ActiveSheet.Range("b46").Value = min(59)      '(Min)
Sheets("Energy Storage, Control Summary").Select
    ActiveSheet.Range("d4").Value = kWh(68)      'Rotary Conv. Spinning
Losses(kWh)
    ActiveSheet.Range("b36").Value = avg(69)      'Battery Bank Voltage (avg)
    ActiveSheet.Range("b37").Value = max(70)      '(Max)
    ActiveSheet.Range("b38").Value = min(71)      '(Min)
        ActiveSheet.Range("b40").Value = avg(85)      'Battery Bank Temp (avg)
        ActiveSheet.Range("b41").Value = max(85)      '(Max)
        ActiveSheet.Range("b42").Value = min(85)      '(Min)
    ActiveSheet.Range("b57").Value = avg(86)      'ESS Shelter Indoor Temp (avg)
    ActiveSheet.Range("b58").Value = max(86)      '(Max)
    ActiveSheet.Range("b59").Value = min(86)      '(Min)
        ActiveSheet.Range("b61").Value = avg(87)      'ESS Shelter Outdoor Temp
(avg)
        ActiveSheet.Range("b62").Value = max(87)      '(Max)
        ActiveSheet.Range("b63").Value = min(87)      '(Min)
Sheets("System Parameters Summary").Select
    ActiveSheet.Range("e17").Value = max(48)      'Max Wind Energy Penetration
End Sub

Sub SOC()      'Battery bank State of Charge (Beginning and Ending)
    Dim begin As Double, ending As Double, LastRow As Integer
    Sheets("Decimals").Select
    LastRow = Application.CountA(ActiveSheet.Range("A:A"))      'Get last used
row of data sheet
    begin = ActiveSheet.Columns(84).Cells(2).Value      '1st SOC value of
month = Beginning SOC
    ending = ActiveSheet.Columns(84).Cells(LastRow).Value      'last SOC value of
month = Ending SOC
    Sheets("Energy Storage, Control Summary").Select
    ActiveSheet.Range("b23").Value = begin
    ActiveSheet.Range("b24").Value = ending
End Sub

Sub charge_discharge()      'Charge, Discharge, Losses, and Efficiency for
battery bank
    Dim i As Integer, j As Integer, LastRow As Integer
    Dim pos As Double, neg As Double, valu As Double
    Dim chg(60 To 72) As Double, disch(60 To 72) As Double
    Dim loss(60 To 72) As Double, effic(60 To 72) As Double
    Sheets("Decimals").Select
    LastRow = Application.CountA(ActiveSheet.Range("A:A"))
    For i = 60 To 72 Step 12      'Col 60 = RC AC kW avg;           Col 72 = RC DC
Amps avg
        pos = 0
        neg = 0
        valu = 0
        chg(i) = 0
        disch(i) = 0
        loss(i) = 0
        effic(i) = 0
        For j = 2 To LastRow
            valu = ActiveSheet.Columns(i).Cells(j).Value
            If valu > 0 Then      'Add up all positive values for Charge
                pos = pos + valu

```

```

        Else                      'Add up all negative values for Discharge
            neg = neg + valu
        End If
    Next j
    chg(i) = pos / 6      'Convert to hours (kw to kWh;  Amps to Ah)
    disch(i) = neg / 6    'same
    loss(i) = chg(i) - disch(i)
    effic(i) = (disch(i) / chg(i))

Next i
Sheets("Energy Storage, Control Summary").Select
ActiveSheet.Range("b26").Value = chg(60)      'RC AC kW
ActiveSheet.Range("b27").Value = disch(60)
ActiveSheet.Range("b28").Value = loss(60)
ActiveSheet.Range("b29").Value = effic(60)
    ActiveSheet.Range("b31").Value = chg(72)      'RC DC Amps
    ActiveSheet.Range("b32").Value = disch(72)
    ActiveSheet.Range("b33").Value = loss(72)
    ActiveSheet.Range("b34").Value = effic(72)
End Sub

Sub state_mode()      'Get amount of time system spent in various states,
modes
    Dim j As Integer, LastRow As Integer
    Dim manual As Integer, mode0 As Integer, model As Integer, mode2 As
Integer, mode3 As Integer
    Dim state0 As Integer, state1A As Integer, state1B As Integer, state2A As
Integer
    Dim state2B As Integer, state3 As Integer, state4 As Integer, state5 As
Integer
    Dim state6 As Integer, state7 As Integer
    Sheets("Raw_Data").Select
    LastRow = Application.CountA(ActiveSheet.Range("A:A"))      'Get last used row
of data sheet
    state0 = 0
    state1A = 0
    state1B = 0
    state2A = 0
    state2B = 0
    state3 = 0
    state4 = 0
    state5 = 0
    state6 = 0
    state7 = 0
    manual = 0
    mode0 = 0
    model = 0
    mode2 = 0
    mode3 = 0
    For j = 2 To LastRow
        Select Case ActiveSheet.Columns(3).Cells(j).Value
            Case "State_0"                  'For each row in "System State"
column, check to see which
                state0 = state0 + 1          'state it's in, then add up how long
(running total)
                Case "State_1A"              'it's been in that state.
                    state1A = state1A + 1
                Case "State_1B"

```

```

        state1B = state1B + 1
    Case "State_2A"
        state0 = state0 + 1
    Case "State_2A"
        state2A = state2A + 1
    Case "State_2B"
        state2B = state2B + 1
    Case "State_3"
        state3 = state3 + 1
    Case "State_4"
        state4 = state4 + 1
    Case "State_5"
        state5 = state5 + 1
    Case "State_6"
        state6 = state6 + 1
    Case "State_7"
        state7 = state7 + 1
    End Select
Select Case ActiveSheet.Columns(2).Cells(j).Value
    Case "Manual_Mode"      'Same deal as above, but with the "System
Mode" column
        manual = manual + 1
    Case "Mode_0"
        mode0 = mode0 + 1
    Case "Mode_1"
        mode1 = mode1 + 1
    Case "Mode_2"
        mode2 = mode2 + 1
    Case "Mode_3"
        mode3 = mode3 + 1
End Select
Next j
Sheets("System Parameters Summary").Select
ActiveSheet.Range("b11").Value = state0
ActiveSheet.Range("b12").Value = state1A
ActiveSheet.Range("b13").Value = state1B
ActiveSheet.Range("b14").Value = state2A
ActiveSheet.Range("b15").Value = state2B
ActiveSheet.Range("b16").Value = state3
ActiveSheet.Range("b17").Value = state4
ActiveSheet.Range("b18").Value = state5
ActiveSheet.Range("b19").Value = state6
ActiveSheet.Range("b20").Value = state7
ActiveSheet.Range("b22").Value = manual
ActiveSheet.Range("b23").Value = mode0
ActiveSheet.Range("b24").Value = mode1
ActiveSheet.Range("b25").Value = mode2
ActiveSheet.Range("b26").Value = mode3
End Sub

Sub dON_dOFF()      'Get the total amount of time the system had some
diesels operating (dON)
    'and how much time all diesels were off (dOFF). Then see
how much wind power
    'was delivered to the village load in each of those two
states.
    Dim j As Integer, i As Integer, LastRow As Integer

```

```

Dim dON As Double, dOFF As Double, kWh(55 To 56) As Double
Dim status1 As Integer, status2 As Integer, status3 As Integer
Dim valu As Double, sum As Double, kW_ON1 As Double, kW_OFF1 As Double
Sheets("Raw_Data").Select
    LastRow = Application.CountA(ActiveSheet.Range("A:A"))      'Get last used
row of data sheet
    dON = 0
    dOFF = 0
    kW_ON1 = 0
    kW_OFF1 = 0
    status1 = 0
    status2 = 0
    status3 = 0
    valu = 0
    For i = 55 To 56
        sum = 0
        For j = 2 To LastRow
            status1 = ActiveSheet.Columns(4).Cells(j).Value      'Diesel 1 status
(ON or OFF)
            status2 = ActiveSheet.Columns(5).Cells(j).Value      'Diesel 2 status
(ON or OFF)
            status3 = ActiveSheet.Columns(6).Cells(j).Value      'Diesel 3 status
(ON or OFF)
            valu = ActiveSheet.Columns(i).Cells(j).Value         'kW delivered (Wind-
to-Load)
            sum = sum + valu
            If status1 = "0" And status2 = "0" And status3 = "0" Then      'All
diesels OFF
                dOFF = dOFF + 1
            Else
                'At least one diesel ON
                dON = dON + 1
            End If
        Next j
        kWh(i) = sum / 6      'Convert kW to kWh
    Next i
    dON = dON / 6          'Convert amount of time to hours
    dOFF = dOFF / 6
Sheets("System Parameters Summary").Select
    ActiveSheet.Range("b8").Value = dON
    ActiveSheet.Range("b9").Value = dOFF
    ActiveSheet.Range("e8").Value = kWh(55)
    ActiveSheet.Range("e9").Value = kWh(56)
    ActiveSheet.Range("e10").Value = "=SUM(R[-2]C:R[-1]C)"
End Sub

Sub data_acq()
    Dim month As Integer, logged As Double, days As Integer, missing As Double
Sheets("Decimals").Select
    month = Left(ActiveSheet.Range("a2").Value, 2)
    Select Case month
        Case 1
            days = 31
        Case 2
            days = 28
        Case 3
            days = 31
        Case 4

```

```

        days = 30
Case 5
    days = 31
Case 6
    days = 30
Case 7
    days = 31
Case 8
    days = 31
Case 9
    days = 30
Case 10
    days = 31
Case 11
    days = 30
Case 12
    days = 31
End Select
logged = Application.CountA(ActiveSheet.Range("A:A"))      '# of used cells
in sheet
Sheets("Data Summary").Select
    ActiveSheet.Range("b2").Value = days * 24      '# of hours in the month
    logged = (logged - 1) / 6                      '# of logged hours in the month (minus
1 for the column header)
    ActiveSheet.Range("b3").Value = logged
    missing = (days * 24) - logged
    ActiveSheet.Range("b4").Value = missing
End Sub

Sub gaps()
    Dim i As Integer, j As Integer
    Dim valu As Double, lastval As Double, diff As Double
    Sheets("Raw_Data").Select
    valu = 0
    lastval = 0
    diff = 0
    i = 3
    Columns("A:A").Select
    Selection.Insert Shift:=xlToRight
    Range("A2").Select
    ActiveCell.FormulaR1C1 = "=VALUE(RC[1])"
    Selection.AutoFill Destination:=Range("A2:A4500"), Type:=xlFillDefault

    Do While Not IsEmpty(ActiveSheet.Columns("a").Cells(i))  'Go through all
cells in the data sheet (there must not be any actual physical gaps in the
data).
        valu = val(ActiveSheet.Columns("a").Cells(i).Value)
        lastval = val(ActiveSheet.Columns("a").Cells(i - 1).Value)
        diff = valu - lastval      'Compare dates of adjacent rows
        If diff >= 0.01388 Then      'if they're at least 20 min. apart, there's
a gap in the data
            ActiveSheet.Columns("a").Cells(i).Select
            Selection.EntireRow.Insert      'if so, insert a new row where
there's a gap
            ActiveSheet.Columns("a").Cells(i).Value = lastval + 0.006944444
0.006944444 is the serial # representing 10 min.

```

```

        ActiveSheet.Columns("b").Cells(i).Value =
ActiveSheet.Columns("a").Cells(i).Value
        ActiveSheet.Columns("b").Cells(i).NumberFormat = "m/d/yy h:mm"      'put
the date that would come next in that new row
        For j = 3 To 105
            ActiveSheet.Columns(j).Cells(i).Value = 0
        Next j
        Rows(i).Select
            Selection.Font.ColorIndex = 3      'Color the "gap" rows red
        End If
        i = i + 1
    Loop
    Columns("A:A").Select
    Selection.Delete Shift:=xlToLeft
    Range("a1").Select
End Sub

Sub missing_segments()
    Dim i As Integer, j As Integer, k As Integer, l As Integer, m As Integer
    Dim start(1 To 50) As Double, ending(1 To 50) As Double
    Dim missing(1 To 1000) As Double, diff As Double      'dimensions of start,
ending, & missing arrays?????
Sheets("Raw_Data").Select
    i = 3
    j = 1      'array of missing values
    l = 1      'array of the segments of missing data
    Do While Not IsEmpty(ActiveSheet.Columns("a").Cells(i))  'should go through
the entire range of data
        If ActiveSheet.Columns("c").Cells(i).Value = "0" Then      'if it's a
missing segment, only col. A&B will be full, C will be empty
            missing(j) = ActiveSheet.Columns("a").Cells(i).Value      'Create array
of all missing segments (their dates)
            j = j + 1
        End If
        i = i + 1
    Loop
    diff = 0
    start(1) = missing(1)      '1st missing value is the start of 1st missing
segment
    For k = 1 To (j - 1)
        diff = missing(k + 1) - missing(k)      'checks to see if missing values are
consecutive (part of same segment of missing data) or not
        If diff > 0.00695 Then          ' 0.00695 is the serial # corresponding
(approx.) to 10min.
            ending(l) = missing(k)      'smaller of the 2 #'s is end of previous
segment
            start(l + 1) = missing(k + 1)      'larger # is the start of the next
segment
            l = l + 1
        End If
    Next k
Sheets("Data Summary").Select
    If j = 1 Then          ' "missing" array is empty if j=1 (i.e. no missing
segments)
        ActiveSheet.Range("b9").FormulaR1C1 = "No Missing Segments"
        Exit Sub
    End If

```



```

        Sheets.Add          '(data for each chart is conveniently grouped
together)
        ActiveSheet.name = "Pie Charts"      'Also creates sheets for all other
charts (except histograms)
        Sheets.Add
        ActiveSheet.name = "Daily Avg Line Graphs"
        Sheets.Add
        ActiveSheet.name = "TS-Wind Speed"
        Sheets.Add
        ActiveSheet.name = "Power Curves"
        Sheets.Add
        ActiveSheet.name = "TS-Freq_Volt Charts"
        Sheets.Add
        ActiveSheet.name = "TS-Battery Charts"
        Sheets.Add
        ActiveSheet.name = "Chart Data"
Dim valu As Double
        Sheets("Wind, Diesel, Load Summary").Range("B3:D3").Copy      'Diesel kWh
delivered
        Sheets("Chart Data").Range("A4").Select
        ActiveSheet.Paste
        Sheets("Wind, Diesel, Load Summary").Range("B15:D15").Copy      'Wind
Turbine kWh delivered
        Sheets("Chart Data").Range("D4").Select
        ActiveSheet.Paste

        Sheets("Energy Storage, Control Summary").Select      'Are Battery Energy
Losses positive (consumed)? or negative (produced)?
        valu = ActiveSheet.Columns("b").Cells(25).Value      'Battery Energy
Losses
        Sheets("Chart Data").Select
        If valu < 0 Then      'If they're negative, they go in "Energy
Production Distribution" chart
            ActiveSheet.Columns("G").Cells(4).Value = Abs(valu)      'Graphed as
a positive value
        Else      'If they're positive, they go in "Energy Consumption
Distribution" chart
            ActiveSheet.Columns("e").Cells(10).Value = valu
        End If

        Sheets("Wind, Diesel, Load Summary").Range("B33:C33").Copy      'Local &
Remote DL kWh consumed
        Sheets("Chart Data").Range("B10").Select
        ActiveSheet.Paste
        Sheets("Wind, Diesel, Load Summary").Range("b43").Copy      'Village load
kWh consumed
        Sheets("Chart Data").Range("A10").Select
        ActiveSheet.Paste
        Sheets("Energy Storage, Control Summary").Range("D4").Copy      'Rotary
Converter spin losses
        Sheets("Chart Data").Range("D10").Select
        ActiveSheet.Paste
        ActiveSheet.Range("b1").Value = "Energy"      'Table 1 Title
        ActiveSheet.Range("c1").Value = "Production"
        ActiveSheet.Range("d1").Value = "Distribution"
        ActiveSheet.Range("e1").Value = "Data"
        ActiveSheet.Range("a3").Value = "Diesel 1"      'Table 1 column headers

```

```

ActiveSheet.Range("b3").Value = "Diesel 2"
ActiveSheet.Range("c3").Value = "Diesel 3"
ActiveSheet.Range("d3").Value = "Wind Turb 1"
ActiveSheet.Range("e3").Value = "Wind Turb 2"
ActiveSheet.Range("f3").Value = "Wind Turb 3"
ActiveSheet.Range("g3").Value = "Energy"           'These two cells look
upside-down in spreadsheet
ActiveSheet.Range("g2").Value = "Losses (neg)"   'so that they'll come out
right as chart labels

ActiveSheet.Range("b7").Value = "Energy"          'Table 2 title
ActiveSheet.Range("c7").Value = "Consumption"
ActiveSheet.Range("d7").Value = "Distribution"
ActiveSheet.Range("e7").Value = "Data"
ActiveSheet.Range("a9").Value = "Village Load"    'Table 2 column headers
ActiveSheet.Range("b9").Value = "Local DL"
ActiveSheet.Range("c9").Value = "Remote DL"
ActiveSheet.Range("d9").Value = "RC Spin"         'These two cells look
upside-down in spreadsheet
ActiveSheet.Range("d8").Value = "Losses"          'so that they'll come out
right as chart labels
ActiveSheet.Range("e9").Value = "Energy"          'These two cells look
upside-down in spreadsheet
ActiveSheet.Range("e8").Value = "Losses (pos)"    'so that they'll come out
right as chart labels

Range("a1:g1, a7:g7").Select
Selection.Font.Bold = True
Selection.Interior.ColorIndex = 15
Selection.Borders(xlEdgeBottom).Weight = xlThin
Selection.Borders(xlEdgeTop).Weight = xlThin
Range("a2:g4, a8:g10").Select
Selection.Font.Bold = False
ActiveSheet.PageSetup.PrintGridlines = True
Cells.Select
Selection.HorizontalAlignment = xlCenter
Cells.EntireColumn.AutoFit
Range("A1").Select
End Sub

Sub daily_averages()      'Creates and formats the "Daily Averages" sheet for
later use with
Sheets.Add                  'some line graphs
ActiveSheet.name = "Daily Averages"
ActiveSheet.Range("a3").Value = "Daily"
ActiveSheet.Range("a4").Value = "Energy"
ActiveSheet.Range("a5").Value = "Production"
ActiveSheet.Range("a6").Value = "And"
ActiveSheet.Range("a7").Value = "Consumption"
ActiveSheet.Range("a8").Value = "Values"
ActiveSheet.Range("a9").Value = "(kWh)"
ActiveSheet.Range("b2").Value = "Day"
ActiveSheet.Range("c1").Value = "Total"
ActiveSheet.Range("c2").Value = "Diesel"
ActiveSheet.Range("d1").Value = "Total"
ActiveSheet.Range("d2").Value = "Wind Turb."
ActiveSheet.Range("e1").Value = "Village"

```

```

ActiveSheet.Range("e2").Value = "Load"
ActiveSheet.Range("f1").Value = "Local"
ActiveSheet.Range("f2").Value = "DL"
ActiveSheet.Range("g1").Value = "Remote"
ActiveSheet.Range("g2").Value = "DL"
ActiveSheet.Range("h1").Value = "RC Spin"
ActiveSheet.Range("h2").Value = "Losses"
ActiveSheet.Range("i1").Value = "Batt. Energy"
ActiveSheet.Range("i2").Value = "Losses"
    ActiveSheet.Range("l3").Value = "Daily"
    ActiveSheet.Range("l4").Value = "Average"
    ActiveSheet.Range("l5").Value = "Wind"
    ActiveSheet.Range("l6").Value = "Speeds"
    ActiveSheet.Range("m1").Value = "Day"
    ActiveSheet.Range("n1").Value = "Turbine 1"
    ActiveSheet.Range("o1").Value = "Turbine 2"
    ActiveSheet.Range("p1").Value = "Turbine 3"
Range("a3:a9, l3:l6").Select
Selection.Font.Bold = True
Selection.Font.Size = 12
Selection.Interior.ColorIndex = 15      'light gray background
Selection.Borders(xlEdgeBottom).Weight = xlMedium
Selection.Borders(xlEdgeRight).Weight = xlMedium
Cells.Select
Selection.NumberFormat = "0.00"
Selection.HorizontalAlignment = xlCenter
Selection.EntireColumn.AutoFit
Range("b:b, m:m").Select
Selection.NumberFormat = "0"
ActiveSheet.PageSetup.PrintGridlines = True
Range("a1").Select
End Sub

Sub dailyavg_calcs()      'Calculates all daily average values for the "Daily
Averages" sheet
    Dim i As Integer, j As Integer, k As Integer, LastRow As Integer
    Dim val As Double, sum As Double, count As Integer
    Dim avg1(1 To 9, 1 To 31) As Double, avg2(1 To 9, 1 To 31) As Double, a(1
To 9) As Integer
    Sheets("Decimals").Select
        a(1) = 24
        a(2) = 45
        a(3) = 57
        a(4) = 49
        a(5) = 50
        a(6) = 68
        a(7) = 60
        a(8) = 35
        a(9) = 42
        For j = 1 To 31      'Up to 31 days/month
            For i = 1 To 9
                sum = 0
                count = 0
                avg1(i, j) = 0
                avg2(i, j) = 0
                For k = 1 To 144      '144 ten minute periods in a day (6*24)

```

```

        If Not IsEmpty(ActiveSheet.Columns(a(i)).Cells(144 * (j - 1) +
k + 1)) Then
            valu = ActiveSheet.Columns(a(i)).Cells(144 * (j - 1) + k +
1).Value
            sum = sum + valu
            count = count + 1           'Should be 144
        End If
    Next k
    If count <> 0 Then
        avg1(i, j) = (sum / 6)      'Get avg kWh from kWh
        avg2(i, j) = (sum / count)   'Get avg wind speed
    Else
        avg1(i, j) = 0             'No data for that day if count = 0
        avg2(i, j) = 0
    End If
Next i
Next j

Sheets("Daily Averages").Select
For j = 1 To 31
    ActiveSheet.Columns("b").Cells(j + 2).Value = j          'Day #
    ActiveSheet.Columns("m").Cells(j + 1).Value = j          'Day #
    ActiveSheet.Columns("c").Cells(j + 2).Value = avg1(1, j)  'Total
Diesel kWh
    ActiveSheet.Columns("d").Cells(j + 2).Value = avg1(2, j)  'Total
Wind Turb kWh
    ActiveSheet.Columns("e").Cells(j + 2).Value = avg1(3, j)
'Village load
    ActiveSheet.Columns("f").Cells(j + 2).Value = avg1(4, j)  'Local
DL
    ActiveSheet.Columns("g").Cells(j + 2).Value = avg1(5, j)  'Remote
DL
    ActiveSheet.Columns("h").Cells(j + 2).Value = avg1(6, j)  'RC
Spin losses
    ActiveSheet.Columns("i").Cells(j + 2).Value = avg1(7, j)
'Battery energy losses
    ActiveSheet.Columns("n").Cells(j + 1).Value = avg2(8, j)
'Turbine 1 avg. wind speed
    ActiveSheet.Columns("o").Cells(j + 1).Value = avg2(9, j)
'Turbine 2 avg. wind speed
    Next j
End Sub

Sub piechart1()      'Energy Production Distribution pie chart (monthly
averages)
    Charts.Add
    ActiveChart.ChartType = xlPie      'or xl3DPie? or xlPieExploded ??
    ActiveChart.SetSourceData Source:=Sheets("Chart Data").Range("A4:G4"),
PlotBy:=xlRows
    ActiveChart.SeriesCollection(1).XValues = "=" & "Chart Data" !R2C1:R3C7"
    ActiveChart.SeriesCollection(1).name = "=" & "Energy Production
Distribution"""
    ActiveChart.Location Where:=xlLocationAsObject, name:="Pie Charts"
                           'or Where:=xlLocationAsNewSheet
    ActiveChart.ApplyDataLabels Type:=xlDataLabelsShowLabelAndPercent,
LegendKey _ :=False, HasLeaderLines:=True

```

```

ActiveChart.SeriesCollection(1).DataLabels.Select
    'do "Selection.Delete" here to get rid of all data labels
ActiveChart.SeriesCollection(1).Points(6).DataLabel.Select
Selection.Delete
ActiveChart.SeriesCollection(1).DataLabels.Select
ActiveChart.SeriesCollection(1).Points(7).DataLabel.Select
Selection.Delete
End Sub

Sub piechart2()      'Energy Consumption Distribution pie chart (monthly
averages)
    Charts.Add
        ActiveChart.ChartType = xlPie      'or xl3DPie or xlPieExploded ??
        ActiveChart.SetSourceData Source:=Sheets("Chart Data").Range("A10:E10"),
PlotBy:=xlRows
        ActiveChart.SeriesCollection(1).XValues = "="'Chart Data'!R8C1:R9C5"
        ActiveChart.SeriesCollection(1).name = "=""Energy Consumption
Distribution"""
        ActiveChart.Location Where:=xlLocationAsObject, name:="Pie Charts"
            'or Where:=xlLocationAsNewSheet
        ActiveChart.ApplyDataLabels Type:=xlDataLabelsShowLabelAndPercent,
LegendKey _
            :=False, HasLeaderLines:=True
End Sub

Sub energy_linegraph()      'Daily Average Energy Production/Consumption line
graph
    Charts.Add
        ActiveChart.ChartType = xlLineMarkers      'or xlLine (for just a line,
not dots or x's)
        ActiveChart.SetSourceData Source:=Sheets("Daily Averages").Range("c:i"),
PlotBy:=xlColumns
        ActiveChart.Location Where:=xlLocationAsObject, name:="Daily Avg Line
Graphs"
        ActiveChart.HasTitle = True
        ActiveChart.ChartTitle.Characters.Text = "Daily Energy
Production/Consumption"
        ActiveChart.Axes(xlCategory, xlPrimary).HasTitle = True
        ActiveChart.Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text =
"Day"
        ActiveChart.Axes(xlValue, xlPrimary).HasTitle = True
        ActiveChart.Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Energy
(kWh)"
        ActiveChart.Axes(xlValue).HasMajorGridlines = True      'Formatting the
new chart
        ActiveChart.Axes(xlCategory).TickLabels.Orientation = xlHorizontal
        ActiveChart.PlotArea.Interior.ColorIndex = xlNone
        ActiveChart.Axes(xlValue).MajorGridlines.Border.ColorIndex = 15  'Light
gray gridlines
End Sub

Sub wind_linegraph()      'Daily Average Wind Speeds (both turbines)
    Charts.Add
        ActiveChart.ChartType = xlLineMarkers      'Line Graph with data points
marked (or xlLine w/o points)
        ActiveChart.SetSourceData Source:=Sheets("Daily Averages").Range("n:p"),
PlotBy:=xlColumns

```

```

ActiveChart.Location Where:=xlLocationAsObject, name:="Daily Avg Line
Graphs"
    ActiveChart.HasTitle = True
    ActiveChart.ChartTitle.Characters.Text = "Daily Average Wind Speeds"
    ActiveChart.Axes(xlCategory, xlPrimary).HasTitle = True
    ActiveChart.Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text =
"Day"
    ActiveChart.Axes(xlValue, xlPrimary).HasTitle = True
    ActiveChart.Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Wind
Speed (mph)"
    ActiveChart.Axes(xlValue).HasMajorGridlines = True           'Formatting the
new chart
    ActiveChart.Axes(xlCategory).TickLabels.Orientation = xlHorizontal
    ActiveChart.PlotArea.Interior.ColorIndex = xlNone
    ActiveChart.Axes(xlValue).MajorGridlines.Border.ColorIndex = 15  'Light
gray gridlines
End Sub

Sub powercurve_wt1()      'WTG1 power curve (power as a function of wind speed)
    Charts.Add
    ActiveChart.ChartType = xlXYScatter        'Scatterplot of ALL recorded data
points
    ActiveChart.SetSourceData
Source:=Sheets("Decimals").Range("AI2:AI4500,AE2:AE4500"), PlotBy:=xlColumns
    ActiveChart.SeriesCollection(1).XValues = "=Decimals!R2C35:R4500C35"
    ActiveChart.SeriesCollection(1).Values = "=Decimals!R2C31:R4500C31"
    ActiveChart.Location Where:=xlLocationAsObject, name:="Power Curves"
    ActiveChart.HasTitle = True
    ActiveChart.ChartTitle.Characters.Text = "Wind Turbine 1 Power Curve"
    ActiveChart.Axes(xlCategory, xlPrimary).HasTitle = True
    ActiveChart.Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "Wind
Speed (mph)"
    ActiveChart.Axes(xlValue, xlPrimary).HasTitle = True
    ActiveChart.Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "WTG1
Power (kW)"
    ActiveChart.PlotArea.Interior.ColorIndex = xlNone
    ActiveChart.Axes(xlValue).MajorGridlines.Border.ColorIndex = 15  'Light gray
gridlines
End Sub

Sub powercurve_wt2()      'WTG2 power curve (power as a function of wind speed)
    Charts.Add
    ActiveChart.ChartType = xlXYScatter        'Scatterplot of ALL recorded
data points
    ActiveChart.SetSourceData
Source:=Sheets("Decimals").Range("AP2:AP4500,AL2:AL4500"), PlotBy:=xlColumns
    ActiveChart.SeriesCollection(1).XValues = "=Decimals!R2C42:R4500C42"
    ActiveChart.SeriesCollection(1).Values = "=Decimals!R2C38:R4500C38"
    ActiveChart.Location Where:=xlLocationAsObject, name:="Power Curves"
    ActiveChart.HasTitle = True
    ActiveChart.ChartTitle.Characters.Text = "Wind Turbine 2 Power Curve"
    ActiveChart.Axes(xlCategory, xlPrimary).HasTitle = True
    ActiveChart.Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text =
"Wind Speed (mph)"
    ActiveChart.Axes(xlValue, xlPrimary).HasTitle = True
    ActiveChart.Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "WTG2
Power (kW)"

```

```

    ActiveChart.PlotArea.Interior.ColorIndex = xlNone
    ActiveChart.Axes(xlValue).MajorGridlines.Border.ColorIndex = 15      'Light
gray gridlines
End Sub

Sub ts_wind1()      'Wind Turbine 1 Time Series Wind Speed chart
    Charts.Add
        ActiveChart.ChartType = xlLine      'Line Graph (w/o points marked)
        ActiveChart.SetSourceData Source:=Sheets("Decimals").Range("Ai:Aj"),
PlotBy:=xlColumns
        ActiveChart.SeriesCollection(1).XValues = "=Decimals!R2C1:R4500C1"
        ActiveChart.SeriesCollection(2).XValues = "=Decimals!R2C1:R4500C1"
        ActiveChart.Location Where:=xlLocationAsObject, name:="TS-Wind Speed"
        ActiveChart.HasTitle = True
        ActiveChart.ChartTitle.Characters.Text = "WTG1 Time Series Wind Speed
Data"
        ActiveChart.Axes(xlCategory, xlPrimary).HasTitle = True
        ActiveChart.Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text =
"Date"
        ActiveChart.Axes(xlValue, xlPrimary).HasTitle = True
        ActiveChart.Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Wind
Speed (mph)"
        ActiveChart.PlotArea.Interior.ColorIndex = xlNone                      'Formatting
the new chart
        ActiveChart.Axes(xlValue).MajorGridlines.Border.ColorIndex = 15      'Light
gray gridlines
        ActiveChart.Axes(xlCategory).TickLabelSpacing = 144                  'Display date
label only once/day
        ActiveChart.Axes(xlCategory).TickLabels.Orientation = xlUpward       'Orient
Date labels vertically
End Sub

Sub ts_wind2()      'Wind Turbine 2 Time Series Wind Speed chart
    Charts.Add
        ActiveChart.ChartType = xlLine      'Line Graph (w/o points marked)
        ActiveChart.SetSourceData Source:=Sheets("Decimals").Range("AP:AQ"),
PlotBy:=xlColumns
        ActiveChart.SeriesCollection(1).XValues = "=Decimals!R2C1:R4500C1"
        ActiveChart.SeriesCollection(2).XValues = "=Decimals!R2C1:R4500C1"
        ActiveChart.Location Where:=xlLocationAsObject, name:="TS-Wind Speed"
        ActiveChart.HasTitle = True
        ActiveChart.ChartTitle.Characters.Text = "WTG2 Time Series Wind Speed
Data"
        ActiveChart.Axes(xlCategory, xlPrimary).HasTitle = True
        ActiveChart.Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text =
"Date"
        ActiveChart.Axes(xlValue, xlPrimary).HasTitle = True
        ActiveChart.Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Wind
Speed (mph)"
        ActiveChart.Axes(xlCategory).TickLabels.Orientation = xlUpward       'Orient
Date labels vertically
        ActiveChart.PlotArea.Interior.ColorIndex = xlNone
        ActiveChart.Axes(xlValue).MajorGridlines.Border.ColorIndex = 15      'Light
gray gridlines
        ActiveChart.Axes(xlCategory).TickLabelSpacing = 144                  'Display
date label only once/day
End Sub

```

```

Sub histogram1()           'Creates and formats the 2 Wind Speed histograms (on
separate worksheets)
    Sheets("Decimals").Select          'Wind Turbine 1
    Application.Run "ATPVBAEN.XLA!Histogram", ActiveSheet.Range("$AI:$AI"), _
                    "Histogram1", , False, False, True, True
        '1st T/F puts bins in order of frequency, most frequent on left,
descending order
        '2nd T/F does Cumulative % stuff
        '3rd T/F creates the chart (false will give you just the frequency
table)
    ActiveSheet.Shapes("Chart 1").ScaleWidth 1.5, msoFalse,
msoScaleFromTopLeft
    ActiveSheet.Shapes("Chart 1").ScaleHeight 2.8, msoFalse,
msoScaleFromTopLeft
    ActiveSheet.ChartObjects("Chart 1").Activate
    ActiveChart.ChartArea.Select
    ActiveChart.Axes(xlCategory).TickLabelSpacing = 20
    ActiveChart.Axes(xlCategory).TickMarkSpacing = 20
    ActiveChart.Legend.Delete      'Don't need a legend, right?
    ActiveChart.ChartTitle.Font.FontStyle = "Bold"
    ActiveChart.ChartTitle.Font.Size = 12
    ActiveChart.ChartTitle.Characters.Text = "Wind Speed Histogram --"
Turbine 1"
    ActiveChart.Axes(xlCategory).AxisTitle.Font.FontStyle = "Bold"
    ActiveChart.Axes(xlCategory).AxisTitle.Font.Size = 10
    ActiveChart.Axes(xlCategory).AxisTitle.Characters.Text = "Wind Speed
(mph)"
    ActiveChart.Axes(xlCategory).TickLabels.Font.Size = 10
    ActiveChart.Axes(xlCategory).TickLabels.Orientation = xlHorizontal
    ActiveChart.Axes(xlValue).AxisTitle.Font.FontStyle = "Bold"
    ActiveChart.Axes(xlValue).AxisTitle.Font.Size = 10
    ActiveChart.Axes(xlValue).TickLabels.Font.Size = 10
Sheets("histogram1").Select
    Range("c1").Select          'Need this for number formatting step (below)
    Columns("A:A").NumberFormat = "0"      'Formats the bin (wind speed)
values--gets rid of all decimals
    ActiveSheet.ChartObjects("Chart 1").Activate
    ActiveChart.PlotArea.Interior.ColorIndex = xlNone
    ActiveChart.PlotArea.Left = 40
    ActiveChart.PlotArea.Width = 345
    ActiveChart.PlotArea.Top = 34
    ActiveChart.PlotArea.Height = 265
End Sub

Sub histogram2()
    Sheets("Decimals").Select          'Wind Turbine 2
    Application.Run "ATPVBAEN.XLA!Histogram", ActiveSheet.Range("$AP:$AP"), _
                    "Histogram2", , False, False, True, True
    ActiveSheet.Shapes("Chart 1").ScaleWidth 1.5, msoFalse,
msoScaleFromTopLeft
    ActiveSheet.Shapes("Chart 1").ScaleHeight 2.8, msoFalse,
msoScaleFromTopLeft
    ActiveSheet.ChartObjects("Chart 1").Activate
    ActiveChart.ChartArea.Select
    ActiveChart.Axes(xlCategory).TickLabelSpacing = 20
    ActiveChart.Axes(xlCategory).TickMarkSpacing = 20

```

```

ActiveChart.Legend.Delete      'Don't need a legend, right?
ActiveChart.ChartTitle.Font.FontStyle = "Bold"
ActiveChart.ChartTitle.Font.Size = 12
ActiveChart.ChartTitle.Characters.Text = "Wind Speed Histogram -- Turbine
2"
ActiveChart.Axes(xlCategory).AxisTitle.Characters.Text = "Wind Speed (mph)"
ActiveChart.Axes(xlCategory).AxisTitle.Font.FontStyle = "Bold"
ActiveChart.Axes(xlCategory).AxisTitle.Font.Size = 10
ActiveChart.Axes(xlCategory).TickLabels.Font.Size = 10
ActiveChart.Axes(xlCategory).TickLabels.Orientation = xlHorizontal
ActiveChart.Axes(xlValue).TickLabels.Font.Size = 10
ActiveChart.Axes(xlValue).AxisTitle.Font.FontStyle = "Bold"
ActiveChart.Axes(xlValue).AxisTitle.Font.Size = 10
Sheets("histogram2").Select
Range("c1").Select
Columns("A:A").NumberFormat = "0"      'Bin number formatting (again)
ActiveSheet.ChartObjects("Chart 1").Activate
ActiveChart.PlotArea.Interior.ColorIndex = xlNone
ActiveChart.PlotArea.Left = 40
ActiveChart.PlotArea.Width = 345
ActiveChart.PlotArea.Top = 34
ActiveChart.PlotArea.Height = 265
End Sub

Sub ts_batt_volt()      'Battery Voltage Time Series line graph
Charts.Add
ActiveChart.ChartType = xlLine      'Line Graph (w/o points marked)
ActiveChart.SetSourceData Source:=Sheets("Decimals").Range("BQ:BS"),
PlotBy:=xlColumns
ActiveChart.SeriesCollection(1).XValues = "=Decimals!R2C1:R4500C1"
ActiveChart.SeriesCollection(2).XValues = "=Decimals!R2C1:R4500C1"
ActiveChart.SeriesCollection(3).XValues = "=Decimals!R2C1:R4500C1"
ActiveChart.Location Where:=xlLocationAsObject, name:="TS-Battery Charts"
ActiveChart.HasTitle = True
ActiveChart.ChartTitle.Characters.Text = "Battery Voltage Time Series
Data"
ActiveChart.Axes(xlCategory, xlPrimary).HasTitle = True
ActiveChart.Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text =
"Date"
ActiveChart.Axes(xlValue, xlPrimary).HasTitle = True
ActiveChart.Axes(xlValue, xlPrimary).AxisTitle.Characters.Text =
"Voltage (V)"
ActiveChart.Axes(xlCategory).TickLabels.Orientation = xlUpward      'Orient
Date labels vertically
ActiveChart.PlotArea.Interior.ColorIndex = xlNone
ActiveChart.Axes(xlValue).MajorGridlines.Border.ColorIndex = 15      'Light
gray gridlines
ActiveChart.Axes(xlCategory).TickLabelSpacing = 144                  'Display date
label only once/day
End Sub

Sub ts_batt_soc_temp()      'Battery SOC & Temp Time Series line graph
Charts.Add
ActiveChart.ChartType = xlLine      'Line Graph (w/o points marked)
ActiveChart.SetSourceData Source:=Sheets("Decimals").Range("CF:CG"),
PlotBy:=xlColumns
ActiveChart.SeriesCollection(1).XValues = "=Decimals!R2C1:R4500C1"

```

```

ActiveChart.SeriesCollection(2).XValues = "=Decimals!R2C1:R4500C1"
ActiveChart.Location Where:=xlLocationAsObject, name:="TS-Battery Charts"
    ActiveChart.HasTitle = True
    ActiveChart.ChartTitle.Characters.Text = "Battery SOC & Temp Time
Series Data"
    ActiveChart.Axes(xlCategory, xlPrimary).HasTitle = True
    ActiveChart.Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text =
"Date"
    ActiveChart.Axes(xlValue, xlPrimary).HasTitle = True
    ActiveChart.Axes(xlValue, xlPrimary).AxisTitle.Characters.Text =
"Battery SOC"
    ActiveChart.ChartArea.Select      'Put Batt. Temp on secondary axis
    ActiveChart.SeriesCollection(2).Select
    ActiveChart.SeriesCollection(2).AxisGroup = 2
    ActiveChart.Axes(xlValue, xlSecondary).HasTitle = True
    ActiveChart.Axes(xlValue, xlSecondary).AxisTitle.Characters.Text =
"Battery Temp"
    ActiveChart.PlotArea.Interior.ColorIndex = xlNone
    ActiveChart.Axes(xlValue).MajorGridlines.Border.ColorIndex = 15
'Light gray gridlines
    ActiveChart.Axes(xlCategory).TickLabels.Orientation = xlUpward      'Orient
Date labels vertically
    ActiveChart.Axes(xlCategory).TickLabelSpacing = 144                  'Display date
label only once/day
End Sub

Sub ts_freq()      'AC Bus Frequency Time Series line graph
    Charts.Add
        ActiveChart.ChartType = xlLine          'Line Graph (w/o points marked)
        ActiveChart.SetSourceData Source:=Sheets("Decimals").Range("R:T"),
PlotBy:=xlColumns
        ActiveChart.SeriesCollection(1).XValues = "=Decimals!R2C1:R4500C1"
        ActiveChart.SeriesCollection(2).XValues = "=Decimals!R2C1:R4500C1"
        ActiveChart.SeriesCollection(3).XValues = "=Decimals!R2C1:R4500C1"
        ActiveChart.Location Where:=xlLocationAsObject, name:="TS-Freq_Volt
Charts"
        ActiveChart.HasTitle = True
        ActiveChart.ChartTitle.Characters.Text = "AC Bus Frequency Time Series
Data"
        ActiveChart.Axes(xlCategory, xlPrimary).HasTitle = True
        ActiveChart.Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text =
"Date"
        ActiveChart.Axes(xlValue, xlPrimary).HasTitle = True
        ActiveChart.Axes(xlValue, xlPrimary).AxisTitle.Characters.Text =
"Frequency"
        ActiveChart.PlotArea.Interior.ColorIndex = xlNone
        ActiveChart.Axes(xlValue).MajorGridlines.Border.ColorIndex = 15
'Light gray gridlines
        ActiveChart.Axes(xlCategory).TickLabelSpacing = 144                  'Display date
label only once/day
        ActiveChart.Axes(xlCategory).TickLabels.Orientation = xlUpward      'Orient
Date labels vertically
End Sub

Sub ts_volt()      'AC Bus Voltage Time Series line graph
    Charts.Add
        ActiveChart.ChartType = xlLine          'Line Graph (w/o points marked)

```

```

    ActiveChart.SetSourceData Source:=Sheets("Decimals").Range("O:Q"),
PlotBy:=xlColumns
    ActiveChart.SeriesCollection(1).XValues = "=Decimals!R2C1:R4500C1"
    ActiveChart.SeriesCollection(2).XValues = "=Decimals!R2C1:R4500C1"
    ActiveChart.SeriesCollection(3).XValues = "=Decimals!R2C1:R4500C1"
    ActiveChart.Location Where:=xlLocationAsObject, name:="TS-Freq_Volt
Charts"
    ActiveChart.HasTitle = True
    ActiveChart.ChartTitle.Characters.Text = "AC Bus Voltage Time Series
Data"
    ActiveChart.Axes(xlCategory, xlPrimary).HasTitle = True
    ActiveChart.Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text =
"Date"
    ActiveChart.Axes(xlValue, xlPrimary).HasTitle = True
    ActiveChart.Axes(xlValue, xlPrimary).AxisTitle.Characters.Text =
"Voltage"
    ActiveChart.PlotArea.Interior.ColorIndex = xlNone
    ActiveChart.Axes(xlValue).MajorGridlines.Border.ColorIndex = 15
'Light gray gridlines
    ActiveChart.Axes(xlCategory).TickLabelSpacing = 144           'Display date
label only once/day
    ActiveChart.Axes(xlCategory).TickLabels.Orientation = xlUpward   'Orient
Date labels vertically
End Sub

Sub daily_avg_format()      'Arranges & formats the Daily Avg Wind Speed and
Energy Production/Consumption charts.
    Sheets("Daily Avg Line Graphs").Select
    ActiveSheet.ChartObjects("Chart 2").Activate           'Daily Avg. Wind
Speed
    ActiveSheet.Shapes("Chart 2").IncrementLeft -249.75
    ActiveSheet.Shapes("Chart 2").IncrementTop 300.75
    ActiveSheet.Shapes("Chart 2").ScaleWidth 1.18, msoFalse,
msoScaleFromTopLeft
    ActiveSheet.Shapes("Chart 2").ScaleHeight 1.44, msoFalse,
msoScaleFromBottomRight
    ActiveChart.Legend.Select
    Selection.Left = 72
    Selection.Width = 290
    Selection.Height = 15
    Selection.Top = 292
    ActiveChart.PlotArea.Select
    Selection.Left = 75
    Selection.Top = 32
    Selection.Width = 308
    Selection.Height = 212
    ActiveChart.Axes(xlCategory).AxisTitle.Select
    Selection.Left = 190
    Selection.Top = 272
    ActiveChart.Axes(xlValue).TickLabels.Font.Size = 10
    ActiveChart.Axes(xlValue).AxisTitle.Font.Size = 10
    ActiveChart.Axes(xlValue).TickLabels.NumberFormat = "0"
    ActiveChart.Axes(xlCategory).TickLabels.Font.Size = 10
    ActiveChart.Axes(xlCategory).TickLabels.NumberFormat = "0"
    ActiveChart.Axes(xlCategory).AxisTitle.Font.Size = 10
    ActiveChart.ChartTitle.Font.Size = 12

```

```

ActiveSheet.ChartObjects("Chart 1").Activate           'Daily Avg. Energy
Production/Consumption
    ActiveSheet.Shapes("Chart 1").IncrementLeft -249#
    ActiveSheet.Shapes("Chart 1").IncrementTop -106.5
    ActiveSheet.Shapes("Chart 1").ScaleWidth 1.18, msoFalse,
msoScaleFromTopLeft
    ActiveSheet.Shapes("Chart 1").ScaleHeight 1.44, msoFalse,
msoScaleFromTopLeft
    ActiveChart.Legend.Select
        Selection.Width = 73
        Selection.Left = 369
        Selection.Top = 10
        Selection.Height = 320
    ActiveChart.PlotArea.Select
        Selection.Left = 75
        Selection.Top = 35
        Selection.Width = 270
        Selection.Height = 215
    ActiveChart.Axes(xlCategory).AxisTitle.Select
        Selection.Top = 370
        Selection.Left = 190
    ActiveChart.Axes(xlValue).TickLabels.Font.Size = 10
    ActiveChart.Axes(xlValue).TickLabels.NumberFormat = "0"
    ActiveChart.Axes(xlValue).AxisTitle.Font.Size = 10
    ActiveChart.Axes(xlCategory).TickLabels.Font.Size = 10
    ActiveChart.Axes(xlCategory).AxisTitle.Font.Size = 10
    ActiveChart.Axes(xlCategory).TickLabels.NumberFormat = "0"
    ActiveChart.ChartTitle.Font.Size = 12
End Sub

Sub wind_ts_format()      'Arranges & formats the 2 wind speed time series
charts
    Sheets("TS-Wind Speed").Select
    ActiveSheet.ChartObjects("Chart 1").Activate
        ActiveSheet.Shapes("Chart 1").IncrementLeft -181.5          'Chart 1 =
Turbine 1
        ActiveSheet.Shapes("Chart 1").IncrementTop -105#            'Chart 2 =
Turbine 2
        ActiveChart.Legend.Select                      'Time Series 1
            Selection.Left = 73
            Selection.Width = 470
            Selection.Height = 14  'ok
            Selection.Top = 290   'ok
    ActiveChart.PlotArea.Select
        Selection.Top = 20
        Selection.Width = 300
        Selection.Height = 182
    ActiveSheet.Shapes("Chart 1").ScaleWidth 1.18, msoFalse,
msoScaleFromTopLeft
        ActiveSheet.Shapes("Chart 1").ScaleHeight 1.44, msoFalse,
msoScaleFromTopLeft
    ActiveChart.Axes(xlCategory).AxisTitle.Select
        Selection.Left = 200
        Selection.Top = 290
    ActiveChart.Axes(xlValue).TickLabels.Font.Size = 10
    ActiveChart.Axes(xlValue).AxisTitle.Font.Size = 10
    ActiveChart.Axes(xlValue).TickLabels.NumberFormat = "0"

```

```

ActiveChart.Axes(xlCategory).TickLabels.Font.Size = 8
ActiveChart.Axes(xlCategory).AxisTitle.Font.Size = 10
ActiveChart.Axes(xlCategory).TickLabels.NumberFormat = "0"
ActiveChart.ChartTitle.Font.Size = 11

ActiveSheet.ChartObjects("Chart 2").Activate      'Time Series 2
ActiveSheet.Shapes("Chart 2").IncrementLeft -118
ActiveSheet.Shapes("Chart 2").IncrementTop 191
ActiveSheet.Shapes("Chart 2").IncrementLeft -63#
ActiveSheet.Shapes("Chart 2").IncrementTop 105#
ActiveSheet.Shapes("Chart 2").ScaleWidth 1.18, msoFalse,
msoScaleFromTopLeft
ActiveSheet.Shapes("Chart 2").ScaleHeight 1.44, msoFalse,
msoScaleFromBottomRight
ActiveChart.Legend.Select
Selection.Left = 73
Selection.Width = 350
Selection.Height = 14
Selection.Top = 290
ActiveChart.PlotArea.Select
Selection.Top = 32
Selection.Width = 300
Selection.Height = 183
ActiveChart.Axes(xlCategory).AxisTitle.Select
Selection.Left = 200
Selection.Top = 290
ActiveChart.Axes(xlValue).TickLabels.Font.Size = 10
ActiveChart.Axes(xlValue).AxisTitle.Font.Size = 10
ActiveChart.Axes(xlValue).TickLabels.NumberFormat = "0"
ActiveChart.Axes(xlCategory).TickLabels.Font.Size = 8
ActiveChart.Axes(xlCategory).AxisTitle.Font.Size = 10
ActiveChart.Axes(xlCategory).TickLabels.NumberFormat = "0"
ActiveChart.ChartTitle.Font.Size = 11
End Sub

```

```

Sub powercurves_format()      'Arranges & formats the 2 power curves
Sheets("Power Curves").Select
ActiveSheet.ChartObjects("Chart 2").Activate      'WTG2 Power Curve
ActiveSheet.Shapes("Chart 2").IncrementLeft -249.75
ActiveSheet.Shapes("Chart 2").IncrementTop 300.75
ActiveSheet.Shapes("Chart 2").ScaleWidth 1.18, msoFalse,
msoScaleFromTopLeft
ActiveSheet.Shapes("Chart 2").ScaleHeight 1.44, msoFalse,
msoScaleFromBottomRight
ActiveChart.Legend.Select      'Don't need a legend, right?
Selection.Delete
ActiveChart.PlotArea.Select
Selection.Left = 70
Selection.Top = 35
Selection.Width = 310
Selection.Height = 215
ActiveChart.Axes(xlCategory).AxisTitle.Select
Selection.Left = 157
Selection.Top = 272
ActiveChart.Axes(xlValue).TickLabels.Font.Size = 10
ActiveChart.Axes(xlValue).AxisTitle.Font.Size = 10
ActiveChart.Axes(xlValue).TickLabels.NumberFormat = "0"

```

```

ActiveChart.Axes(xlCategory).TickLabels.Font.Size = 8
ActiveChart.Axes(xlCategory).AxisTitle.Font.Size = 10
ActiveChart.Axes(xlCategory).TickLabels.NumberFormat = "0"
ActiveChart.ChartTitle.Font.Size = 12
ActiveChart.ChartTitle.Font.Bold = True

ActiveSheet.ChartObjects("Chart 1").Activate           'WTG1 Power Curve
    ActiveSheet.Shapes("Chart 1").IncrementLeft -249#
    ActiveSheet.Shapes("Chart 1").IncrementTop -106.5
    ActiveSheet.Shapes("Chart 1").ScaleWidth 1.18, msoFalse,
msoScaleFromTopLeft
    ActiveSheet.Shapes("Chart 1").ScaleHeight 1.44, msoFalse,
msoScaleFromTopLeft
    ActiveChart.Legend.Select
        Selection.Delete
    ActiveChart.PlotArea.Select
        Selection.Left = 80
        Selection.Top = 35
        Selection.Width = 310
        Selection.Height = 215
    ActiveChart.Axes(xlCategory).AxisTitle.Select
        Selection.Left = 157
        Selection.Top = 272
    ActiveChart.Axes(xlValue).TickLabels.Font.Size = 10
    ActiveChart.Axes(xlValue).AxisTitle.Font.Size = 10
    ActiveChart.Axes(xlValue).TickLabels.NumberFormat = "0"
        ActiveChart.Axes(xlCategory).TickLabels.Font.Size = 8
        ActiveChart.Axes(xlCategory).AxisTitle.Font.Size = 10
        ActiveChart.Axes(xlCategory).TickLabels.NumberFormat = "0"
    ActiveChart.ChartTitle.Font.Size = 12
    ActiveChart.ChartTitle.Font.Bold = True
End Sub

Sub freq_volt_format()      'Arranges & formats the AC Bus Frequency and
Voltage Time Series charts
    Sheets("TS-Freq_Volt Charts").Select
    ActiveSheet.ChartObjects("Chart 2").Activate           'ts_volt (AC Bus
Voltage)
    ActiveSheet.Shapes("Chart 2").IncrementLeft -249.75
    ActiveSheet.Shapes("Chart 2").IncrementTop 300.75
    ActiveSheet.Shapes("Chart 2").ScaleWidth 1.18, msoFalse,
msoScaleFromTopLeft
    ActiveSheet.Shapes("Chart 2").ScaleHeight 1.44, msoFalse,
msoScaleFromBottomRight
    ActiveChart.Legend.Select
        Selection.Left = 55
        Selection.Width = 370
        Selection.Height = 16
        Selection.Top = 292
    ActiveChart.PlotArea.Select
        Selection.Left = 75
        Selection.Top = 35
        Selection.Width = 300
        Selection.Height = 225
    ActiveChart.Axes(xlCategory).AxisTitle.Select
        Selection.Left = 190
        Selection.Top = 272

```

```

ActiveChart.Axes(xlValue).TickLabels.Font.Size = 10
ActiveChart.Axes(xlValue).TickLabels.NumberFormat = "0"
ActiveChart.Axes(xlValue).AxisTitle.Font.Size = 10
ActiveChart.Axes(xlCategory).TickLabels.Font.Size = 8
ActiveChart.Axes(xlCategory).AxisTitle.Font.Size = 10
ActiveChart.Axes(xlCategory).TickLabels.NumberFormat = "0"
ActiveChart.ChartTitle.Font.Size = 12

ActiveSheet.ChartObjects("Chart 1").Activate           'ts_freq (AC Bus
Frequency)
    ActiveSheet.Shapes("Chart 1").IncrementLeft -249#
    ActiveSheet.Shapes("Chart 1").IncrementTop -106.5
    ActiveSheet.Shapes("Chart 1").ScaleWidth 1.18, msoFalse,
msoScaleFromTopLeft
    ActiveSheet.Shapes("Chart 1").ScaleHeight 1.44, msoFalse,
msoScaleFromTopLeft
    ActiveChart.Legend.Select
        Selection.Height = 17
        Selection.Left = 55
        Selection.Width = 373
        Selection.Top = 292
    ActiveChart.PlotArea.Select
        Selection.Left = 75
        Selection.Top = 35
        Selection.Width = 300
        Selection.Height = 225
    ActiveChart.Axes(xlCategory).AxisTitle.Select
        Selection.Left = 190
        Selection.Top = 272
    ActiveChart.Axes(xlValue).TickLabels.Font.Size = 10
    ActiveChart.Axes(xlValue).TickLabels.NumberFormat = "0"
    ActiveChart.Axes(xlValue).AxisTitle.Font.Size = 10
    ActiveChart.Axes(xlCategory).TickLabels.Font.Size = 8
    ActiveChart.Axes(xlCategory).AxisTitle.Font.Size = 10
    ActiveChart.Axes(xlCategory).TickLabels.NumberFormat = "0"
    ActiveChart.ChartTitle.Font.Size = 12
End Sub

Sub battery_format()      'Arranges & formats the Battery charts (SOC & Temp,
Voltage)
    Sheets("TS-Battery Charts").Select
        ActiveSheet.ChartObjects("Chart 2").Activate           'Battery SOC & Temp
        ActiveSheet.Shapes("Chart 2").IncrementLeft -249.75
        ActiveSheet.Shapes("Chart 2").IncrementTop 300.75
        ActiveSheet.Shapes("Chart 2").ScaleWidth 1.18, msoFalse,
msoScaleFromTopLeft
        ActiveSheet.Shapes("Chart 2").ScaleHeight 1.44, msoFalse,
msoScaleFromBottomRight
        ActiveChart.Legend.Select
            Selection.Left = 102
            Selection.Width = 230
            Selection.Height = 14
            Selection.Top = 292
    ActiveChart.PlotArea.Select
        Selection.Left = 75
        Selection.Top = 35
        Selection.Width = 275

```

```

    Selection.Height = 267
ActiveChart.Axes(xlCategory).AxisTitle.Select
    Selection.Left = 227
    Selection.Top = 272
ActiveChart.Axes(xlValue).TickLabels.Font.Size = 10
ActiveChart.Axes(xlValue).AxisTitle.Font.Size = 10
ActiveChart.Axes(xlValue).TickLabels.NumberFormat = "0"
ActiveChart.Axes(xlValue, xlSecondary).TickLabels.Font.Size = 10
ActiveChart.Axes(xlValue, xlSecondary).TickLabels.NumberFormat = "0"
ActiveChart.Axes(xlValue, xlSecondary).AxisTitle.Font.Size = 10
ActiveChart.Axes(xlCategory).TickLabels.Font.Size = 8
ActiveChart.Axes(xlCategory).AxisTitle.Font.Size = 10
ActiveChart.Axes(xlCategory).TickLabels.NumberFormat = "0"
ActiveChart.ChartTitle.Font.Size = 12

ActiveSheet.ChartObjects("Chart 1").Activate      'Battery Voltage
    ActiveSheet.Shapes("Chart 1").IncrementLeft -249#
    ActiveSheet.Shapes("Chart 1").IncrementTop -106.5
    ActiveSheet.Shapes("Chart 1").ScaleWidth 1.18, msoFalse,
msoScaleFromTopLeft
    ActiveSheet.Shapes("Chart 1").ScaleHeight 1.44, msoFalse,
msoScaleFromTopLeft
    ActiveChart.Legend.Select
        Selection.Height = 17
        Selection.Left = 55
        Selection.Width = 373
        Selection.Top = 288
    ActiveChart.PlotArea.Select
        Selection.Left = 80
        Selection.Top = 35
        Selection.Width = 300
        Selection.Height = 255
    ActiveChart.Axes(xlCategory).AxisTitle.Select
        Selection.Left = 223
        Selection.Top = 272
    ActiveChart.Axes(xlValue).TickLabels.Font.Size = 10
    ActiveChart.Axes(xlValue).TickLabels.NumberFormat = "0"
    ActiveChart.Axes(xlValue).AxisTitle.Font.Size = 10
    ActiveChart.Axes(xlCategory).TickLabels.Font.Size = 8
    ActiveChart.Axes(xlCategory).AxisTitle.Font.Size = 10
    ActiveChart.Axes(xlCategory).TickLabels.NumberFormat = "0"
    ActiveChart.ChartTitle.Font.Size = 12
End Sub

Sub pie_charts_format()      'Arranges & formats the 2 pie charts
Sheets("Pie Charts").Select
    ActiveSheet.ChartObjects("Chart 2").Activate      'Energy Consumption Pie
Chart
    ActiveSheet.Shapes("Chart 2").IncrementLeft -183#
    ActiveSheet.Shapes("Chart 2").IncrementTop 301.5
    ActiveSheet.Shapes("Chart 2").ScaleWidth 1.18, msoFalse,
msoScaleFromTopLeft
    ActiveSheet.Shapes("Chart 2").ScaleHeight 1.44, msoFalse,
msoScaleFromBottomRight
    ActiveChart.Legend.Select
        Selection.Width = 50
        Selection.Left = 373

```

```

Selection.Height = 146
ActiveChart.Legend.Font.Size = 8
ActiveChart.SeriesCollection(1).Select
ActiveChart.PlotArea.Select
Selection.Left = 71
Selection.Top = 40
Selection.Width = 258
Selection.Height = 258
ActiveChart.SeriesCollection(1).DataLabels.Font.Size = 8
ActiveChart.SeriesCollection(1).Points(5).DataLabel.Select      'Data label
placements may
Selection.Left = 74                                         'need some
work
Selection.Top = 42
ActiveChart.SeriesCollection(1).Points(3).DataLabel.Select
Selection.Left = 175
Selection.Top = 263
ActiveChart.SeriesCollection(1).Points(1).DataLabel.Select
Selection.Left = 279
Selection.Top = 49
ActiveChart.SeriesCollection(1).Points(2).DataLabel.Select
Selection.Left = 330
Selection.Top = 198
ActiveChart.ChartTitle.Font.Size = 12
ActiveChart.ChartTitle.Font.Bold = True

ActiveSheet.ChartObjects("Chart 1").Activate      'Energy Distribution Pie
Chart
ActiveSheet.Shapes("Chart 1").IncrementLeft -182.25
ActiveSheet.Shapes("Chart 1").IncrementTop -106.5
ActiveSheet.Shapes("Chart 1").ScaleWidth 1.18, msoFalse,
msoScaleFromTopLeft
ActiveSheet.Shapes("Chart 1").ScaleHeight 1.44, msoFalse,
msoScaleFromTopLeft
ActiveChart.Legend.Select
Selection.Width = 46
Selection.Left = 376
Selection.Height = 194
Selection.Top = 83
ActiveChart.Legend.Font.Size = 8
ActiveChart.PlotArea.Select
Selection.Left = 71
Selection.Top = 40
Selection.Width = 258
Selection.Height = 258
ActiveChart.SeriesCollection(1).DataLabels.Font.Size = 8
ActiveChart.SeriesCollection(1).Points(3).DataLabel.Select      'Data
label placements may
Selection.Left = 175                                         'need
some work.
Selection.Top = 263
ActiveChart.SeriesCollection(1).Points(5).DataLabel.Select
Selection.Left = 76
Selection.Top = 42
ActiveChart.SeriesCollection(1).Points(1).DataLabel.Select
Selection.Left = 274
Selection.Top = 42

```

```

ActiveChart.SeriesCollection(1).Points(2).DataLabel.Select
Selection.Left = 330
Selection.Top = 189
ActiveChart.ChartTitle.Font.Size = 12
ActiveChart.ChartTitle.Font.Bold = True
End Sub

Sub sheet_placement()      'Organizes worksheet tabs in an appropriate
manner.
    Sheets("Raw_Data").Move Before:=Sheets(1)
    Sheets("Alarms").Move Before:=Sheets(2)
    Sheets("ESS_Alarms").Move Before:=Sheets(3)
    Sheets("Decimals").Move Before:=Sheets(4)
    Sheets("Wind, Diesel, Load Summary").Move Before:=Sheets(5)
    Sheets("Energy Storage, Control Summary").Move Before:=Sheets(6)
    Sheets("System Parameters Summary").Move Before:=Sheets(7)
    Sheets("Data Summary").Move Before:=Sheets(8)
    Sheets("Chart Data").Move Before:=Sheets(9)
    Sheets("Daily Averages").Move Before:=Sheets(10)
    Sheets("Daily Avg Line Graphs").Move Before:=Sheets(11)
    Sheets("Pie Charts").Move Before:=Sheets(12)
    Sheets("Power Curves").Move Before:=Sheets(13)
    Sheets("TS-Wind Speed").Move Before:=Sheets(14)
    Sheets("Histogram1").Move Before:=Sheets(15)
    Sheets("Histogram2").Move Before:=Sheets(16)
    Sheets("TS-Freq_Volt Charts").Move Before:=Sheets(17)
    Sheets("TS-Battery Charts").Move Before:=Sheets(18)
    Sheets("Raw_Data").Select
End Sub

```

